

**UNIVERSIDAD AUTONOMA DE MADRID**

**ESCUELA POLITECNICA SUPERIOR**



**Grado en Ingeniería de Tecnologías y Servicios de  
Telecomunicación**

## **TRABAJO FIN DE GRADO**

**ANÁLISIS DE LA EVOLUCIÓN, EN NÚMERO Y TAMAÑO, DE  
LESIONES DE PIEL EN ZONAS AMPLIAS DEL CUERPO**

**Juan Antonio Álvarez Castillo  
Tutor: Dr. Jesús Bescós Cano**

**JUNIO 2020**



# **ANÁLISIS DE LA EVOLUCIÓN, EN NÚMERO Y TAMAÑO, DE LESIONES DE PIEL EN ZONAS AMPLIAS DEL CUERPO**

**AUTOR: Juan Antonio Álvarez Castillo**  
**TUTOR: Dr. Jesús Bescós Cano**



**Video Procesing and Understanding Lab**  
**Escuela Politécnica Superior**  
**Universidad Autónoma de Madrid**  
**JUNIO 2020**



## Resumen (castellano)

El objetivo principal de este trabajo de fin de grado es la detección en zonas amplias del cuerpo de manera automática de posibles cambios en lesiones dermatológicas, ya sea aparición, aumento de tamaño o cambio de color de estas. En la actualidad, no se dispone de dicha tecnología, lo cual genera un problema ya que, en el momento del análisis visual por parte del facultativo, cabe la posibilidad de que se pasen por alto ciertas lesiones, además de que el tiempo empleado en la revisión de los cambios es mayor. Este trabajo intenta dar solución a estos problemas mediante el apoyo al proceso de detección del cambio en las zonas amplias del cuerpo. El proceso de detección se ha llevado a cabo mediante el tratamiento de las dos imágenes que se quieren comparar, las cuales, pasan por un proceso de selección de la zona de interés; una adaptación de estas con el fin de poder compararlas de la manera más precisa posible; y, por último, la comparación propiamente dicha de las imágenes y la visualización de los resultados obtenidos. Para el desarrollo de este proceso se ha realizado un estudio de las técnicas existentes y, finalmente, de las empleadas en el desarrollo del algoritmo, explicando en qué se basan dichas técnicas y cómo se han aplicado para obtener el resultado deseado.

Para la realización de las pruebas y visualización de resultados, ante la imposibilidad de contar con el *Dataset* inicialmente previsto, se generó el mismo formado por diez imágenes, simulando las condiciones reales que se puedan dar en una consulta dermatológica, donde pudiéramos comprobar el funcionamiento de cada etapa del algoritmo para comprobar de esta manera su efectividad.

## Palabras clave (castellano)

Segmentación, detección, contorno, lesión, variación de la lesión, crecimiento de la lesión, homografía, comparación, dermatología.

## Abstract

The main objective of this Bachelor's Thesis is the automatic detection in wide areas of the body of possible changes in dermatological injuries, whether their appearance, or if they increase in size or changed in color. At the moment, such technology is not available, which is a problem as certain injuries may be overlooked at the time of visual analysis and the time taken to review changes might be greater. This thesis tries to solve these problems by supporting the process of detecting changes in large areas of the body. The detection process has been carried out by processing two images to be compared, which have gone through a selection process of the area of interest; an adaptation of these in order to be able to compare them as accurately as possible; and, finally, the actual comparison of the images and the display of the results obtained. For the development of this process, a study of the existing techniques and a final description of those used in the development of the algorithm has been done, as well as explaining what they are based on and how they have been applied to obtain the desired result.

To carry out the tests and visualize the results and due to the impossibility of not having the initial planned one, a *Dataset* was generated consisting of ten images, which simulate the real conditions that can occur in a dermatological practice, where we could check the operation of each step of the algorithm to verify its effectiveness.

## Keywords

Segmentation, detection, contour, injury, injury variation, injury growth, homography, comparison, dermatology.



## ***Agradecimientos***

En primer lugar, me gustaría agradecerle a mi familia, y en especial a mi madre, a mi padre y a mi hermano, no solo por preocuparse por mí durante todo el desarrollo de este trabajo, sino durante toda la carrera, animándome, apoyándome y haciéndome ver que rendirse es la solución más fácil pero nunca la mejor. Gracias a esos “empujoncitos” y a vuestra paciencia he podido llegar hasta aquí. Muchas gracias de corazón.

Por otra parte, muchas gracias a mi tutor, el Dr. Jesús Bescós, el cual seguro que muchas veces ha debido tener mucha paciencia conmigo debido a mis incesantes correos en las últimas etapas del trabajo. Gracias también por tus ideas y consejos en las tutorías, lo creas o no, hacías que una bombilla se convirtiera en una lámpara llena de luces.

Por último, estas palabras van para ti: Sé que hubieras sido el primero en haberte leído este trabajo, sé que hubieras sido el primero en celebrarlo con su “nietecico”, me gusta imaginar ese abrazo que me hubieras dado y ese pin que me hubieras puesto. Por todo, solo puedo darte las gracias desde aquí y dejarte este párrafo para demostrarte que siempre has estado muy presente desde que nací, y aún sin estar lo estás ... Siempre has sido y serás de las personas más grandes, más buenas, más perseverantes, más entregadas a su familia y amigos, y el mejor hijo, padre, abuelo y hermano que nadie podría tener. A la orden siempre mi comandante.



# INDICE DE CONTENIDOS

Resumen	i
Abstract	ii
Agradecimientos	iv
1 Introducción	1
1.1 Motivación	1
1.2 Objetivos	2
1.3 Organización de la memoria	2
2 Antecedentes y tecnologías de base	3
2.1 Introducción	3
2.2 Antecedentes	3
2.3 Técnicas aplicadas	3
2.3.1 Segmentación de regiones de piel	4
2.3.2 Extracción de contornos	6
2.3.3 Cálculo de homografías entre dos imágenes	7
2.3.4 Igualación de color	8
2.3.5 Comparación entre dos imágenes	8
3 Especificación de requisitos y diseño funcional de la aplicación	10
3.1 Introducción	10
3.2 Especificación de requisitos	10
3.3 Diseño funcional de la aplicación	11
3.3.1 Indicaciones sobre la captura de imágenes	11
3.3.2 Modo de ejecución del algoritmo	12
4 Desarrollo	15
4.1 Introducción	15
4.2 Descripción general del algoritmo	15
4.3 Funciones desarrolladas	19
4.3.1 Operaciones previas	19
4.3.2 Segmentador	19
4.3.3 Detector de contornos	21
4.3.4 Homografía	22
4.3.5 Igualación de color	23
4.3.6 Comparación	24
4.3.7 Visualización de resultados	25
4.4 Parámetros de diseño del algoritmo	26
5 Integración, pruebas y resultados	34
5.1 Dataset	34
5.2 Resultados visuales	34
6 Conclusiones y trabajo futuro	38
6.1 Conclusiones	38
6.2 Trabajo futuro	39
Referencias	XL

## INDICE DE FIGURAS

1.1: Ejemplo imágenes comparación y resultado .....	1
2.2: Ejemplo de resultado del segmentador basado en regiones .....	5
2.3: Ejemplo de resultado del segmentador basado en umbral .....	6
2.4: Ejemplo contornos activos. ....	7
2.5: Ejemplo de puntos característicos. ....	8
2.6: Diferencia de dos imágenes y reresentación .....	9
3.1: Ejemplo de como capturar imágenes .....	11
3.2: Como extraer propiedades de una imagen .....	13
3.3: Variacion de archivo para la ejecución del algoritmo .....	14
3.4: Imagen de ejemplo sobre el resultado del algoritmo .....	14
4.1: Diagrama de bloques del algoritmo .....	15
4.2: Imágenes sobre las que se describirá de manera general el algoritmo .....	16
4.3: Imágenes segmentadas .....	16
4.4: Contornos imágenes a comparar . ....	17
4.5: Puntos característicos e Imagen resultado de la homografía .....	17
4.6: Imagen resultado de la ecualización de color .....	18
4.7: Imagen resultado de la comparación .....	18
4.8: Imágenes que se utilizaran para explicar las funciones desarrolladas .....	19
4.9: Ejemplo sobre la orientación de la imagen .....	19
4.10: Ejemplo de resultado del segmentador de color .....	20
4.11: Zonas con falsas segmentaciones .....	20
4.12: Distintos resultados de la etapa de extracción de contornos.....	22
4.13: Imágenes a comparar y sus zonas de búsqueda de puntos característicos.....	23
4.14: Representación puntos característicos, relación entre los mismos y resultado de la homografía .....	23
4.15: Imágenes sobre las que se aplica la ecualización de color y resultado .....	24
4.16: Resultado de la diferencia entre dos imágenes y su representación .....	25
4.17: Ejemplo de imagen donde se muestran los cambios encuadrados. ....	26
4.18: Ejemplo de resultado que obtendría el usuario del algoritmo. ....	26
4.19: Imágenes para la evaluación de sensibilidad del contorno. ....	27
4.20: Resultado de la diferencia entre imágenes con una reducción de la zona de interés del 0,01% .....	28
4.21: Resultado de la diferencia entre imágenes con una reducción de la zona de interés del 0,1%. ....	28
4.22: Resultado de la diferencia entre imágenes con una reducción de la zona de interés del 0,3%. ....	29
4.23: Resultado de la diferencia entre imágenes con una reducción de la zona de interés del 0,5%. ....	29
4.24: Resultado de la diferencia entre imágenes con una sensibilidad de 1 y la diferencia en el contorno eliminada.....	30
4.25: Resultado de la diferencia entre imágenes con una sensibilidad de 10 y la diferencia en el contorno eliminada.....	31

4.26: Resultado de la diferencia entre imágenes con una sensibilidad de 20 y la diferencia en el contorno eliminada.....	31
4.27: Resultado de la diferencia entre imágenes con una sensibilidad de 1 y la diferencia en el contorno eliminada.....	32
4.28: Resultado de la sensibilidad del algoritmo con sensibilidad 1 para otra fotografía ..	32
5.1: Resultados del algoritmo con una sensibilidad de 25 y una reducción de la zona de interés del 0,4%. ....	37

# 1 Introducción

---

## 1.1 Motivación

Un problema a día de hoy en cuanto a la detección de lesiones dermatológicas se refiere (ya sean lunares, manchas u otros defectos que podemos encontrar en la piel) es la detección de las mismas de una manera manual, siendo el propio usuario el que se encargue de comparar dos imágenes, extrayendo así las nuevas lesiones que hayan podido surgir con el paso del tiempo, es decir, lo que el usuario busca son las diferencias entre ambas imágenes para así ver cómo han variado las mismas con el paso del tiempo. Por tanto, en este trabajo se propone la detección automática de dichos cambios en la piel donde, haciendo uso de dos imágenes del mismo paciente tomadas en distintos intervalos temporales, seamos capaces de detectar los cambios citados anteriormente de una manera automática sin que el usuario tenga que analizar por sí mismo las fotografías. Esta implementación permitirá a sus usuarios un considerable ahorro de tiempo, así como, en una implementación ideal, evitar el fallo humano que se puede producir al ser las imágenes comparadas de una manera manual. A continuación, se muestra de una manera más gráfica la motivación del trabajo, donde veremos las dos imágenes que un usuario compararía y el resultado de donde se encuentran las diferencias entre ambas:

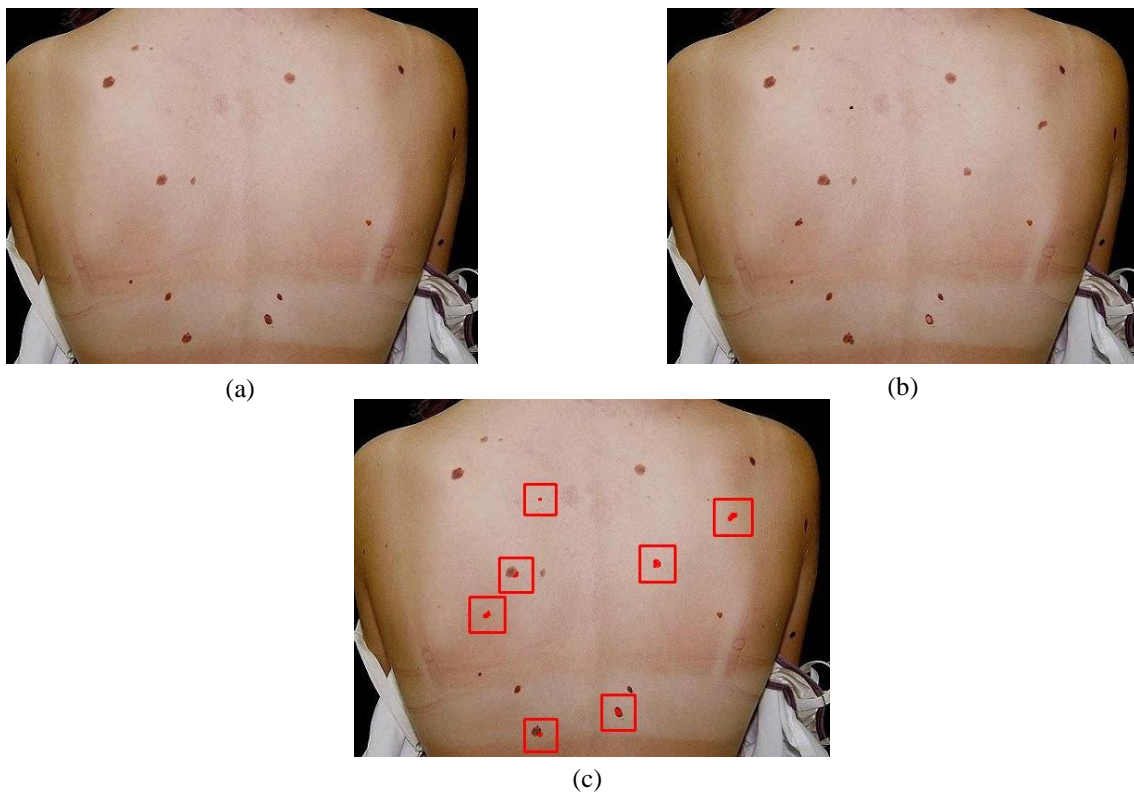


Figura 1.1: Imágenes a comparar (a) y (b). Resultado de comparación (c). Imagen (a) extraída de [14]

## **1.2 Objetivos**

El objetivo de este trabajo es identificar las nuevas lesiones de la piel, o bien, las variaciones de estas, que hayan podido surgir en el transcurso de tiempo de la captura entre las dos imágenes. Para ello habrá que realizar un pre procesamiento de la imagen, esto es, tratar la imagen de tal manera que se pueda conseguir separar la región de interés (en este trabajo la piel) del resto de la imagen, intentar paliar las diferencias en el color de la piel que hayan podido surgir a lo largo del tiempo, así como la distinta colocación que pueda haber con las imágenes que tenemos que comparar debido a que, por ejemplo, el usuario las haya tomado en un ángulo diferente. Una vez hecho esto ya se podrán comparar ambas imágenes para extraer las potenciales diferencias que tenemos entre ambas.

Los puntos que se van a tratar son los siguientes:

1. El estudio del estado del arte, donde se mostrará la tecnología ya existente con respecto a este trabajo.
2. Toda la información referente a los algoritmos y métodos existentes que se han utilizado para la realización de este trabajo.
3. La implementación utilizada para el trabajo y la algoritmia de este.
4. Resultados obtenidos en la detección de los nuevos cambios de la piel.

## **1.3 Organización de la memoria**

La memoria de este trabajo se va a componer de los siguientes capítulos.

- Capítulo 1 Introducción: motivación y objetivos.
- Capítulo 2 Antecedentes y tecnología de base.
- Capítulo 3 Diseño: Algoritmia existente y tipos.
- Capítulo 4 Desarrollo: Algoritmia.
- Capítulo 5 Pruebas y resultados.
- Capítulo 6 Conclusiones y trabajo futuro.
- Bibliografía.

## 2 Antecedentes y tecnologías de base

---

### 2.1 Introducción

En este apartado se hará referencia en primer lugar los antecedentes relacionados a este trabajo, es decir, una serie de aplicaciones ya existentes en la actualidad las cuales, ayudan a conseguir el fin último de este trabajo que es ser capaces de detectar los cambios que hayan podido surgir en alguna zona de la piel. Posteriormente se realizará una revisión de las técnicas aplicadas para el desarrollo del algoritmo que se ha aplicado para la detección de nuevas lesiones.

### 2.2 Antecedentes

En la actualidad ya existe *software* (el programa) que permiten realizar la detección de nuevas lesiones dermatológicas de una manera cómoda. Una empresa destacada es *Canfield Scientific* la cual se encarga en desarrollar *software* con aplicaciones médicas basada en fotografías, así como la venta de *hardware* (cámaras, trípodes...) para complementarlo. Relacionado con este trabajo podemos destacar también *Mirror DermaGraphix body mapping software*, el cual, no detecta automáticamente las lesiones, sino que permite introducir manualmente las mismas en base a una fotografía de una zona amplia del cuerpo, y, enlazar la lesión con el histórico de la misma que se tomó en un periodo temporal anterior. Por otro lado, también tenemos la aplicación *SkinIO* desarrollada por *SkinIO HQ* donde puedes registrar distintos pacientes y dentro de cada uno se pueden detectar nuevas lesiones de una manera tanto manual como automática, así como en el caso anterior, enlazar cada lesión a un histórico que permita ver de una manera más rápida y clara la evolución de las lesiones. Ambas aplicaciones mencionadas en esta sección son de pago, costando unos 4.700€ y 150€ anuales respectivamente.

Otras aplicaciones existentes hoy en día son aquellas que miden la evolución de las lesiones obtenidas mediante un dermatoscopio o cualquier otra herramienta para detectar cambios, así como la evolución de lesiones concretas, ya existentes y conocidas ya que este producto permite estudiar las lesiones que se han detectado visualmente pudiendo haber fallo humano en dicha detección visual.

### 2.3 Técnicas aplicadas

En este apartado se desarrollará de una manera lo mas clara y concisa posible las técnicas necesarias para el desarrollo del algoritmo. En cada uno de los subapartados se mostrarán tanto las técnicas existentes que no fueron útiles para el desarrollo del algoritmo, así como la técnica seleccionada finalmente para el desarrollo de este.

### 2.3.1 Segmentación de regiones de piel

Esta técnica lo que permitirá será extraer la zona de interés de la imagen eliminando el resto. Para la extracción de la zona de interés se ha barajado el uso de distintos tipos de segmentadores los cuales se van a intentar explicar de una manera clara en este apartado, qué tipo hemos seleccionado y el porqué de dicha elección. A continuación, se van a exponer los dos métodos estudiados para la extracción de la zona de interés:

- 1- **Segmentador basado en bordes:** Este tipo de segmentador lo que detecta son los bordes que hay en una imagen, dando como resultado una imagen en la cual podemos distinguir únicamente los bordes. El problema que se ha encontrado con este tipo de segmentador en cuanto al desarrollo de nuestro trabajo se refiere, es que no nos permite seleccionar únicamente la zona que queremos comparar, sino que detectaría todos los bordes que hay en la imagen, esto es, por ejemplo, el borde entre el fondo y la espalda, los distintos bordes que originan los lunares y demás lesiones, entre otros bordes que se puedan encontrar en la imagen, por lo que no nos es de especial utilidad (ver figura 2.1).

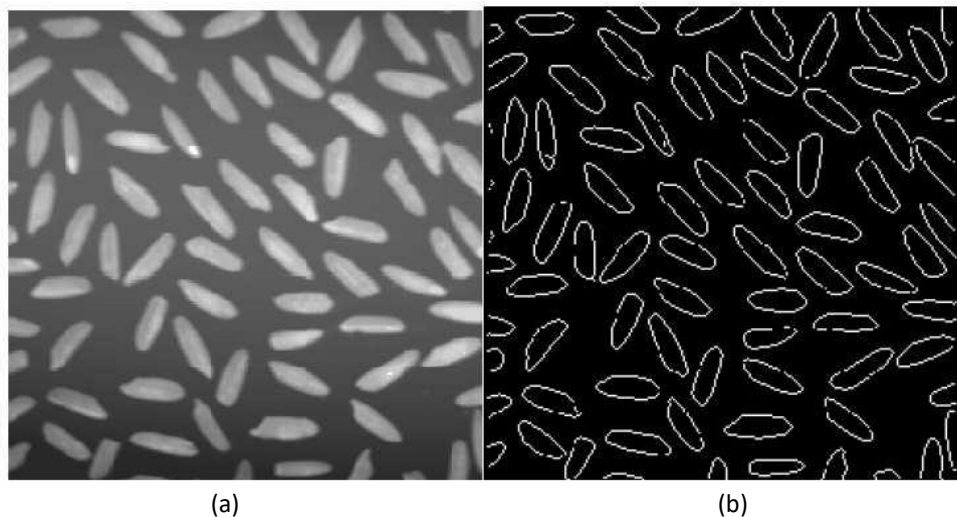


Figura 2.1: Ejemplo de resultado del segmentador basado en bordes. Imagen (a) a segmentar, imagen (b) resultado de la segmentación. Extraído de [29]

- 2- **Segmentador basado en regiones:** La función que realiza este tipo de segmentador es seleccionar las regiones completas que cumplan las características especificadas en el programa, es decir, este tipo de segmentador divide la imagen en zonas que son similares entre sí. Dentro de este tipo de segmentador se han estudiado dos métodos para la extracción de la zona de interés:

**2.1. Método de crecimiento de regiones:** La función de este método es localizar zonas que cumplan una serie de características partiendo de unos puntos de la imagen llamados semillas, definidas por el usuario, las cuales van aumentando su tamaño hasta que se deje de cumplir la condición establecida lo que daría como resultado una región. Por ejemplo, una condición podría ser que cada píxel se asigne a una región si la diferencia con el valor de la semilla es menor a diez. Los problemas que encontramos con este tipo de segmentador son los siguientes: El primero es que requiere un alto coste computacional, esto es, requiere que el ordenador donde se ejecute utilice muchos recursos para llevarlo a cabo y, el segundo, y más importante, es que todos los parámetros necesarios para seleccionar la región que nos interesa hay que definirlos en función de cada imagen concreta, estudiando su composición y por ende, no tendríamos una funcionalidad global que funcionara en un amplio abanico de casos, lo cual no es de utilidad para el trabajo. Por ejemplo, los parámetros utilizados para conseguir la Figura 2.2 no necesariamente funcionarían para otra imagen con distintas características.

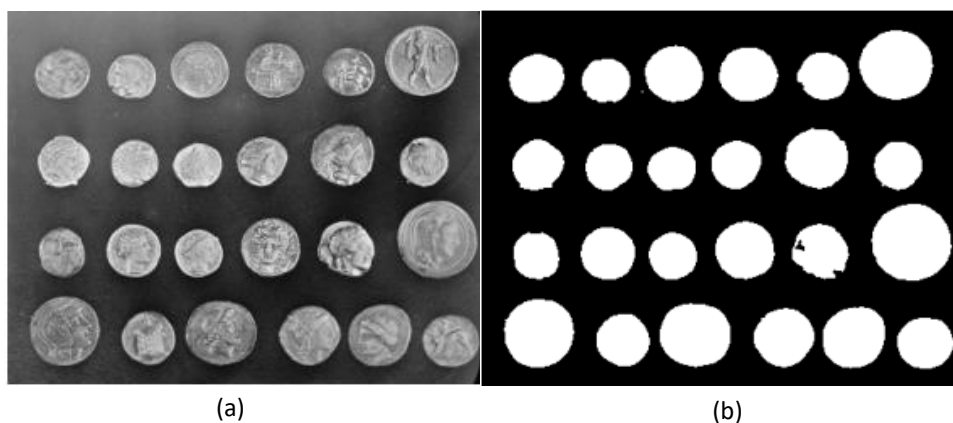


Figura 2.2: Ejemplo de resultado del segmentador basado en regiones. Imagen (a) a segmentar, imagen (b) resultado de la segmentación. Extraído de [26]

**2.2. Método del valor umbral (umbralización):** Este tipo de segmentador es el más conveniente para abordar el problema al que se intenta dar solución en este trabajo. Este segmentador selecciona aquellas regiones que cumplan una serie de características que nosotros hemos establecido previamente como, por ejemplo, el color. A diferencia del método de crecimiento de regiones, que proporciona una solución concreta para un problema concreto, este segmentador nos permitirá abordar el problema de una manera más general, pudiendo definir una serie de características comunes que se cumplen en todas las imágenes generalizando así este paso, es decir,



de una manera más clara, en vez de buscar regiones concretas, buscamos zonas con características concretas como lo es el color de la piel implementando así un segmentador de color, el cual será el utilizado en el desarrollo del algoritmo (ver figura 2.3).



Figura 2.3: Ejemplo de resultado del segmentador basado en umbral (de color). Imagen (a) a segmentar, imagen (b) resultado de la segmentación. Imagen (a) extraída de [14]

### **2.3.2 Extracción de contornos**

Esta técnica nos permitirá extraer el contorno de la imagen segmentada, el cual se utilizará para distintos fines como se explicará en el capítulo 4.2.3. Para la extracción de contornos se barajaron tres métodos:

1. **Detector de bordes:** Este detector permite extraer todos los bordes existentes en la imagen, no obstante, detectaría bordes que no nos interesarían para el desarrollo de la aplicación como, por ejemplo, aquellos bordes existentes entre la lesión y la piel, entre el cuerpo y el fondo de la imagen entre otros, por lo que, no sería posible saber cuál es el borde que coincidiera con el contorno del cuerpo. Este detector tiene el mismo principio de funcionamiento que el segmentador basado en bordes donde básicamente, la diferencia radicaría en la utilidad que se le da (ver figura 2.1).
2. **Detección de contorno en base al cambio de color:** El segundo método que se barajo fue detectar cuando la imagen varía de negro a otro tono, lo que nos permitiría extraer el contorno, pero al tener que comprar cada punto de la imagen, generaría un alto coste computacional, hecho que ralentizará en exceso la aplicación por lo que no se llegó a implementar.
3. **Detector por contornos activos:** Con este método se obtuvieron resultados muy satisfactorios y el cual se ha implementado para el desarrollo del algoritmo. Este detector lo podríamos definir como un elemento elástico que se ajusta al contorno que deseamos detectar. Este elemento elástico comienza en el límite de la imagen, encogiéndose poco a poco y ajustándose a la zona

del contorno que queremos detectar obteniendo así el contorno deseado (ver figura 2.4).

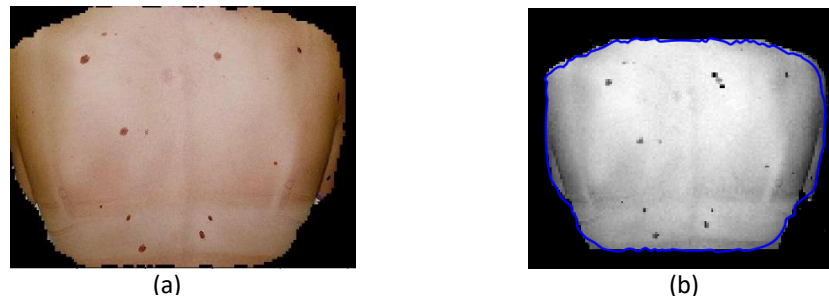


Figura 2.4: Ejemplo de extracción de contornos mediante el método de contornos activos. Imagen (a) a la que se le detecta el contorno. Imagen (b) resultado de la detección.

### 2.3.3 Cálculo de homografías entre dos imágenes

Se define como homografía a la transformación proyectiva de una imagen con respecto a otra. Este proceso consiste en detectar puntos característicos (puntos que tienen unas propiedades concretas) entre ambas imágenes y en función de estos, realizar las operaciones necesarias para obtener ambas imágenes como si se hubieran capturado con el mismo ángulo de visión y profundidad. El paso más importante del cálculo de homografías entre dos imágenes es la detección de los citados puntos característicos ya que, en función de la fiabilidad y precisión de estos, se consigue una transformación lo más fiel posible.

Para la realización de este trabajo hemos analizado tres métodos para detectarlos: **SIFT (Scale-invariant feature transform)**, **SURF (Speeded-Up Robust Features)** y por último **ORB (Oriented fast and Rotative Brief)**. Estos métodos en esencia realizan lo mismo, detectan y describen los puntos característicos de la imagen, es decir, en qué lugar de la imagen se encuentran y que característica tienen estos. Si los descriptores de dos imágenes coinciden, los puntos correspondientes a los dichos descriptores serán el mismo punto en sendas imágenes (ver figura 2.5), los cuales, se utilizarán para transformar una imagen con respecto a la otra. La diferencia entre los métodos reside en la capacidad de detección y en la rapidez, siendo **SIFT** el método que mejor detecta los puntos, pero por el contrario es notablemente más lento que los otros dos métodos. En cuanto a **SURF** y **ORB** se obtienen resultados similares, pero finalmente nos decantamos por el método **ORB** ya que es ligeramente más rápido que **SURF** y se obtienen unos resultados similares entre ambos, y dado que las imágenes generalmente son de buena calidad no se hace necesario un detector más preciso como lo es **SIFT**.

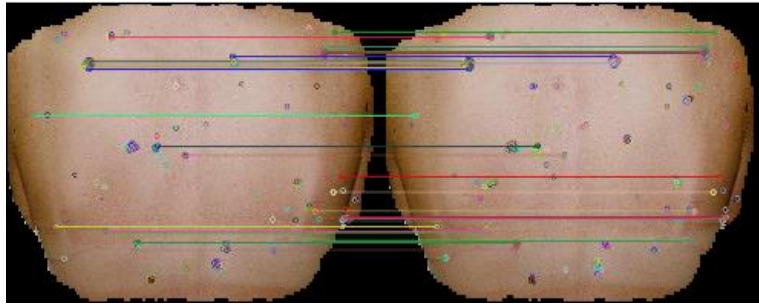


Figura 2.5: Ejemplo de representación de puntos característicos de dos imágenes y la relación entre los mismos.

### 2.3.4 Igualación de color

En este apartado se presenta una revisión de algunas técnicas de tratamiento de imagen existentes, las cuales, permiten cambiar ciertos parámetros de las imágenes con el principal fin de mejorar ciertos aspectos de la imagen. Algunos de estos métodos son:

1. **Ecualización de histograma:** El histograma es una representación gráfica de la distribución de los distintos tonos de la imagen, es decir, si por ejemplo tenemos una imagen en blanco y negro, qué cantidad tenemos de negro, de blanco, de gris, etc. Con este método podemos modificar el rango de valores en el que se mueven los canales de una imagen, para obtener un rango más amplio en los mismos, permitiendo así obtener un mayor contraste.
2. **Promedio de imágenes:** Método que permite suavizar cambios abruptos en las imágenes, como, por ejemplo, el ruido que se genera al realizar una foto nocturna que no la permite ver con claridad.
3. **Métodos basados en píxeles:** Método que varía el valor que tienen los píxeles de la imagen con un fin determinado. Este fue el método seleccionado para el desarrollo del algoritmo ya que el problema que se planteaba era la diferencia de color de manera uniforme a lo largo de la imagen, es decir, si una persona ha estado expuesta a la luz solar, el color de su piel habrá variado de manera uniforme, entonces, tomando como referencia la primera imagen capturada, comprobamos la diferencia de color que hay entre las dos imágenes, y variamos los valores de la segunda, para así, igualar el color lo máximo posible. La implementación de esta técnica se especificará en el Capítulo 4.

### 2.3.5 Comparación entre dos imágenes

Para realizar la comparación entre imágenes, el método estudiado fue la diferencia entre imágenes (restar dos imágenes). Esto es debido a que es una manera clara de extraer las diferencias entre imágenes con la que se obtiene un buen resultado,

siempre y cuando, se hayan aplicado de la manera correcta los métodos descritos anteriormente como bien se verá en el punto 4. Este método lo que nos proporciona es una visión clara de donde se sitúan las diferencias, partiendo de la base que aquellas zonas que se hayan mantenido invariantes con el paso del tiempo tendrán una diferencia prácticamente nula, y las zonas que hayan variado tendrán una diferencia notable. Una vez localizadas, lo único que hay que hacer es representarlas en la imagen original (ver figura 2.6)



Figura 2.6: Imagen de diferencias (a) y representación de estas (b)

## 3 Especificación de requisitos y diseño funcional de la aplicación

---

### 3.1 Introducción

En este capítulo se expondrán los objetivos de la aplicación desde un punto de vista funcional. En primer lugar, se explicará la especificación de requisitos, es decir, que aspectos requiere y espera el usuario del detector. Posteriormente, se especificará en qué condiciones y cómo se deben tomar las fotografías a comparar para que el algoritmo funcione correctamente y con el menor error posible. Por último, se explicará brevemente como un usuario podría ejecutar el algoritmo en el estado del desarrollo en el que se encuentra.

### 3.2 Especificación de requisitos

En este apartado se especificarán una serie de requisitos planteados por un usuario para que el algoritmo le sea de utilidad, los cuales, denominaremos **requisitos indispensables**. Por otro lado, se especificarán aquellos requisitos que extenderán a los requisitos indispensables y que aportarían una mejor interactividad, usabilidad y utilidad. A estos últimos los denominaremos **requisitos adicionales**.

En cuanto a **requisitos indispensables** destacaremos los siguientes:

- El usuario debe ser capaz de detectar los nuevos cambios que han podido surgir entre dos imágenes de una manera rápida, clara e intuitiva, mostrando las dos imágenes que se han comparado y una tercera donde se resaltan las diferencias.
- Se debe poder detectar no solo la aparición de nuevas alteraciones en la piel sino también cambios de color o tamaño de las ya existentes.
- Es de especial utilidad no solo saber dónde se encuentran los cambios en la piel, sino que también, ser capaces de observar el área donde se han producido dichos cambios.
- En cuanto a la captura de imágenes, aunque lleven asociadas ciertos requisitos que se especificarán en el apartado 3.3, deben de poder realizarse sin atender en exceso a factores como la iluminación (se requerirá buena iluminación, pero no una iluminación de un estudio de fotografía). Por otro lado, el usuario podrá utilizar el dispositivo de captura de imágenes que más se adapte a sus necesidades, sin necesidad de utilizar un dispositivo de captura de imágenes específico. Este punto tiene como objetivo emplear el mínimo tiempo posible en la captura de imágenes.

En cuanto a **requisitos adicionales** se refiere destacaremos los siguientes:

- El usuario debe ser capaz de seleccionar él mismo la región de la piel que desea comparar para así, poder ser más selectivos en cuanto a la zona donde desee observar las diferencias, ya que, no necesariamente le podría interesar analizar toda la región de piel capturada por la cámara.
- Se debe de disponer de una interfaz clara e intuitiva que permita al usuario realizar la comparación entre imágenes de una manera rápida, seleccionando directamente las dos imágenes y ejecutando la aplicación.
- La aplicación debe mantener un registro de cambios donde el usuario pueda consultar dichos cambios que se produjeron en sesiones anteriores, así como acceder a todas las imágenes que han sido tomadas al paciente en las distintas sesiones de una manera fácil y rápida.

### ***3.3 Diseño funcional de la aplicación***

En este apartado se detallarán una serie de indicaciones para la captura de las imágenes con el fin de obtener los mejores resultados al ejecutar el algoritmo y, por otro lado, se propondrá un procedimiento para ser capaces de utilizar el algoritmo implementado. Cabe destacar en este apartado que, para el desarrollo del algoritmo únicamente se han tenido en cuenta los requisitos indispensables citados en el apartado 3.2.

#### ***3.3.1 Indicaciones sobre la captura de imágenes***

Para el correcto funcionamiento de esta aplicación hay que tener en cuenta unos detalles para que el funcionamiento de esta sea lo mejor posible. Para la detección de cambios en la zona de la espalda, que es la zona para la que se ha optimizado el algoritmo, las imágenes deben ser capturadas preferiblemente con un fondo blanco, negro o con un color totalmente distinto al color que puede tener la piel, y evitando, en la medida de lo posible, que se vean los brazos en la imagen, por ejemplo, colocando estos por delante del pecho (ver figura 3.1).



Figura 3.1: Ejemplo (a) (b) (c) de cómo se deben tomar las imágenes. (a) extraída de [10], (b) extraída de [28] y (c) extraída de [2]

Por otro lado, sería buena idea tener dos marcas en el suelo, la primera donde se deberá colocar el paciente, y la segunda, donde se colocará el usuario que va a realizar la fotografía, con el fin de que la captura de ambas imágenes se asemeje lo máximo posible en cuanto a distancia y perspectiva se refiere. La aplicación corregirá las pequeñas variaciones que puede haber entre ambas fotografías capturadas. También, en cuanto a la toma de las imágenes, habrá que tener en cuenta que estas se deben tomar con una buena iluminación y evitando que los límites de la zona donde queramos comprobar los posibles cambios no coincidan con los límites de la fotografía.

En siguiente lugar, las imágenes se deberán tomar desde un inicio en la misma posición, es decir, si la primera imagen se toma con la cámara en posición horizontal, las sucesivas se deberán tomar en posición horizontal, y si se toman en vertical, las sucesivas imágenes se deberán capturar en vertical. Este punto es independiente para cada paciente, es decir, si tenemos dos pacientes, y al primero le capturamos la primera imagen en posición vertical las sucesivas imágenes se deberán capturar en posición vertical, pero, si al segundo paciente, por la razón que sea, las imágenes conviene más tomarlas en posición horizontal todas han de ser tomadas en horizontal. Esto se debe a que las imágenes, por lo general, no tienen el mismo tamaño en cuanto ancho y alto se refiere por lo que, al ejecutar el algoritmo, habrá un error en las dimensiones de la imagen a comparar.

### ***3.3.2 Modo de ejecución del algoritmo***

Para ejecutar el algoritmo con el que se obtendrán la comparación de imágenes se deberá realizar los siguientes pasos.

- 1- Se deberán tener en el ordenador las imágenes que se deseen comparar que habrán sido capturadas como se explica en el punto anterior.
- 2- Instalar Python 3.8 en el ordenador. Se podrá obtener desde el siguiente enlace: <https://www.python.org/downloads/>
- 3- Modificar el código para incluir la ruta de las dos imágenes a comparar. Para ello: pulsar encima de la imagen y pulsar **botón derecho** sobre la misma y buscar **propiedades**. Con esto aparecerá la siguiente pantalla y se deberán tener en cuenta los campos resaltados:

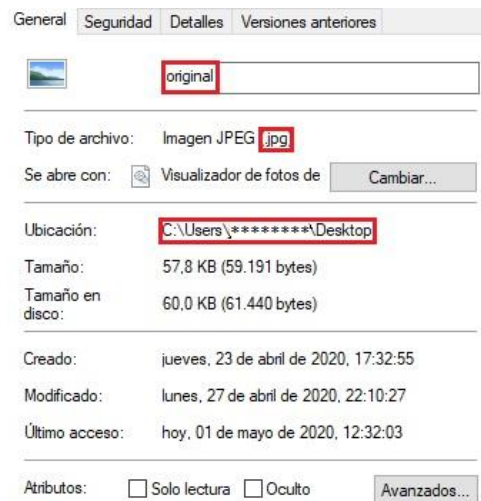
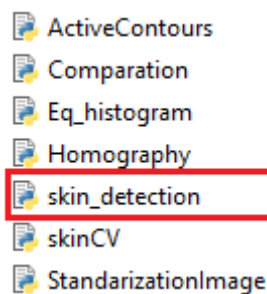


Figura 3.2: Imagen de ejemplo para extraer la información

- 4- Con los campos anteriormente resaltados deberemos crear la ruta, esto es algo del tipo: **C:\Users\\*\*\*\*\*\Desktop\original.jpg**

Repetir los pasos 2 y 3 con la imagen con la que queremos comparar.

- 5- Abrir el archivo **skin\_detection** y añadir las rutas donde se indica. Habrá que tener en cuenta que la ruta de “imagen\_actual” e “imagen representación actual” es la misma. Esto lo podrá hacer un usuario poco experimentado instalando un editor de texto como puede ser el notepad o el sublime text (ver Figura 3.3). Guardar los cambios realizados.



(a)



```

from StandarizationImage import StandardImage
from skinCV import SkinDetector
from ActiveContours import Segmentator
from Homography import homography
from Eq_histogram import EqHistogram
from Comparation import comparation
import cv2

imagen_antigua = r"C:\Users\*****\Desktop\Fotos\Pacientel\2019.jpg"
imagen_actual = r"C:\Users\*****\Desktop\Fotos\Pacientel\2020-8-VM.jpg"
imagen_actual_representación = r"C:\Users\*****\Desktop\Fotos\Pacientel\2020-8-VM.jpg"

standard_antigua = StandardImage(imagen_antigua)
orientation_antigua = standard_antigua.flags()
standard_actual = StandardImage(imagen_actual)
orientation_actual = standard_actual.flags()
standard_actual_representacion = StandardImage(imagen_actual)
orientation_representacion = standard_actual.flags()

```

(b)

Figura 3.3: Archivo a editar (a). Imagen de ejemplo sobre colocación de la ruta(b).

- 6- Por último, el usuario deberá dar doble clic encima del archivo **skin\_detection** para la obtención del resultado (ver Figura 3.4).



Figura 3.4: Imagen de ejemplo del resultado que se obtiene al ejecutar el algoritmo.

## 4 Desarrollo

---

### 4.1 Introducción

En esta sección se presentará, en primer lugar, una descripción general del algoritmo. Posteriormente en detalle todos los pasos seguidos para obtener las diferencias entre las dos imágenes a comparar. Por último, se mostrará la influencia de una serie de parámetros del algoritmo de los cuales depende la precisión del comparador a la hora de comparar dos imágenes.

### 4.2 Descripción general del algoritmo

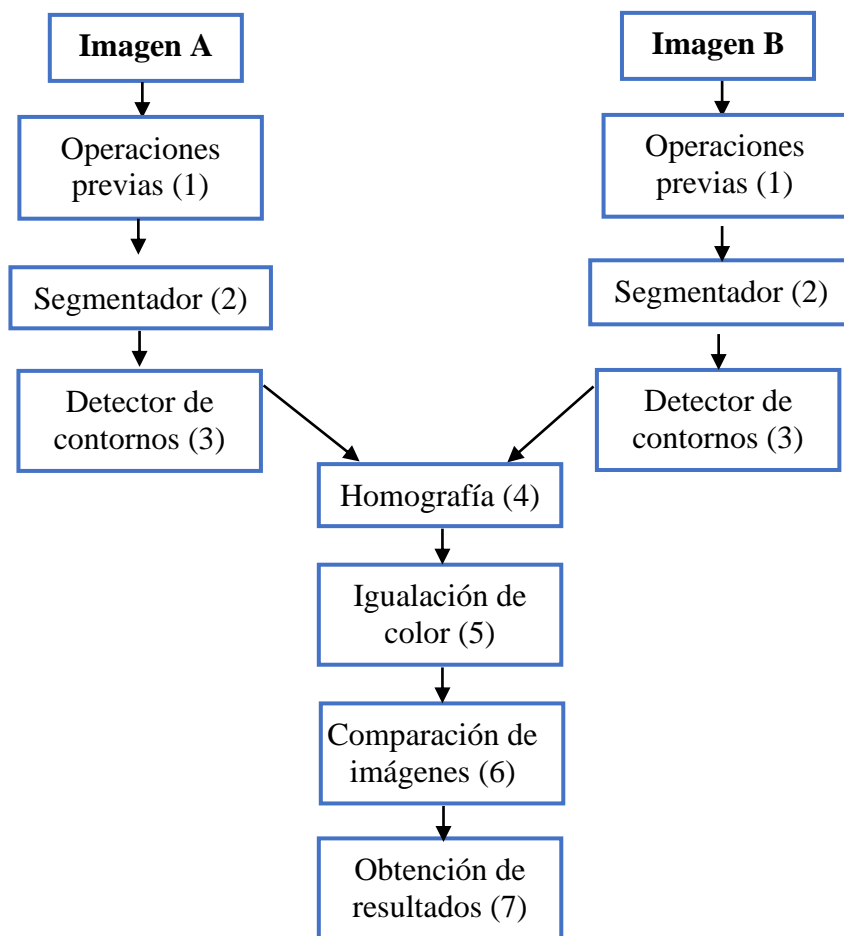


Figura 4.1: Diagrama de bloques del algoritmo

En este apartado se explicará de manera general las técnicas que se han aplicado, así como terminología propia de las mismas que pueden aparecer a lo largo del trabajo.

El punto de partida son las dos imágenes que queremos comprobar si han tenido cambios a lo largo del tiempo (Imagen A e Imagen B) como se puede ver en la figura 4.2.



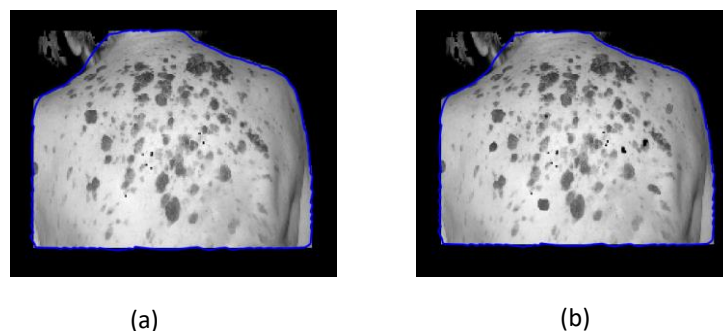
Figura 4.2: Imágenes a comparar (a) y (b). Imagen (a) extraída de [28]

En primer lugar, se realizan una serie de **operaciones previas (1)** las cuales comprueban la posición de la imagen y la sitúan siempre en horizontal para que sea visualizada correctamente. Posteriormente, una vez tenemos las imágenes en la posición correcta, deberemos seleccionar únicamente la zona de interés, en nuestro caso, la piel. Esta selección se realizará con un proceso llamado **segmentación (2)**.



Figura 4.3: Imágenes segmentadas

Dado que puede suceder que ciertas zonas de piel el algoritmo no considere piel, se hace necesario **detectar el contorno (3)** de las imágenes anteriores para ser capaces de extraer la zona de piel, pero proveniente de la imagen original para eliminar esos “errores en la segmentación” como podemos ver en la figura 4.4.



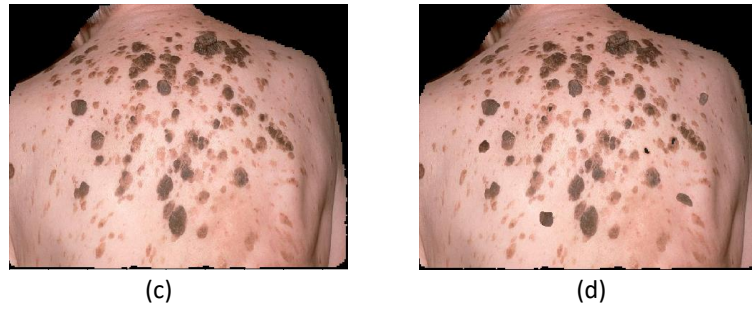


Figura 4.4: Contornos imágenes a comparar (a) y (b). Resultado de la extracción zona de interés (c) y (d) procedentes de los contornos (a) y (b) respectivamente.

Una vez se ha extraído la zona de interés en ambas imágenes el siguiente paso es hacer la **homografía (4)**. Este paso consiste en hallar los puntos característicos de ambas imágenes los cuales servirán de referencia para colocar (mover, rotar, agrandar ...) ambas imágenes en la misma posición si estas hubiera sido tomadas en distinto lugar, distancia y con un ángulo de colocación de la cámara ligeramente distinto. De una manera más clara, si sabes que puntos son los mismos en ambas imágenes solo tienes que hacerlos coincidir en la misma posición variando solo una imagen de las dos, así, como resultado, obtendremos la imagen con todas las transformaciones necesarias como podemos ver en la figura 4.5.

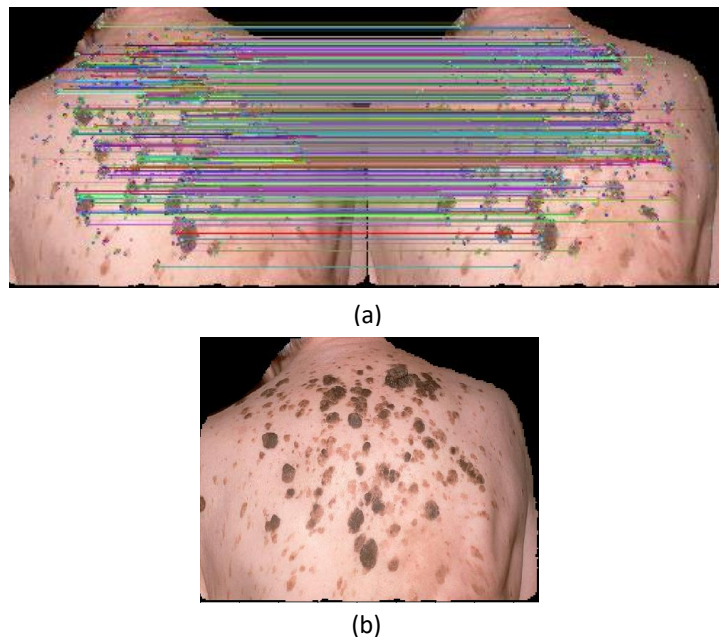


Figura 4.5: Puntos característicos (a) e Imagen resultado de la homografía (b)

Por último, antes de realizar la comparación, debemos **igualar el color (5)** entre las imágenes. Pongamos el supuesto que el sujeto de las imágenes se hace una foto en abril, y la siguiente foto que se realiza es en septiembre, si en ese tiempo, por algún motivo, su

ha variado, la comparación, debido al metodo que se aplíca en el algoritmo, el cual se explicará en profundidad en el apartado 4.3.6, hará dicha comparación mas complicada. Como resultado obtendremos una imagen con dicha operación ya realizada (ver figura 4.6).

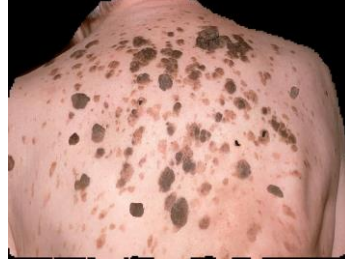


Figura 4.6: Imagen resultado de la ecualización de color

En este último paso del algoritmo, en el que, una vez se ha eliminado en la medida de lo posible todas aquellas zonas que no son necesarias para realizar la comparación (**segmentación**), en las que se ha variado la perspectiva (**homografía**) y el tono (**igualación de color**) para intentar simular que han sido tomadas en las mismas condiciones se procede a **compararlas (6)**. Este último paso dará como **resultado (7)** las dos imágenes a comparar y las diferencias halladas entre las mismas.



Figura 4.7: Imagen resultado de la comparación



### 4.3 Funciones desarrolladas

En este apartado se va a explicar detalladamente cada paso llevado a cabo para el desarrollo del algoritmo. Todos los ejemplos de este apartado se harán en base a las fotografías mostradas en la figura 4.8.



Figura 4.8: Imágenes que se utilizaran para explicar las funciones desarrolladas. Imagen (a) extraída de [18]

#### 4.3.1 Operaciones previas

El punto de partida del algoritmo son las dos imágenes que se quieren comparar. Esta primera etapa rota, si es necesario, las imágenes para que ambas estén en posición horizontal.

Para ello lee la posición en la que se encontraba la cámara en el momento de tomar la fotografía aprovechando que al capturarla se generan unas etiquetas dentro del archivo de imagen, las cuales siguen el estándar *EXIF* (*Exchangeable image file format*) donde, una de estas etiquetas especifica la orientación. Si esta etiqueta tiene valor 1, no será necesario rotar la imagen, Si tiene valor 8 la imagen se rotará 90°, si tiene valor 3 se rotará 180° y si por último tiene valor 6 se rotará 270° todas ellas en sentido contrario a las agujas del reloj (ver Figura 4.9).

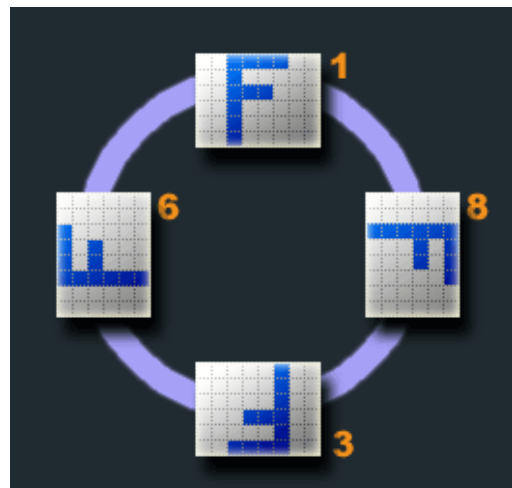


Figura 4.9: Ejemplo sobre la orientación de la imagen. Figura extraída de [7]

#### 4.3.2 Segmentador

Como se ha explicado en el apartado 4.2, esta etapa es necesaria para eliminar todas aquellas zonas que no resultan de interés para realizar la comparación. Por

otro lado, y conforme se ha explicado en el Capítulo 2, en esta etapa se va a implementar con un segmentador por color.

La funcionalidad que realiza esta parte del algoritmo se lleva a cabo de la siguiente manera: En primer lugar, hay que tener en cuenta que el color de las imágenes con las que estamos tratando se consigue por la combinación, en la proporción adecuada, de los tres colores primarios, siendo estos colores el rojo, el verde y el azul (RGB por sus siglas en inglés). En las imágenes digitales dichos colores se presentan en tres capas (canales) donde cada una representa uno de los mismos en las proporciones adecuadas y, combinando los tres, formamos la imagen a color.

Una vez sabemos esto, lo que definimos en esta etapa del algoritmo es qué rango de valores de cada componente, combinándolos entre sí, forman el color que podamos considerar piel, obteniendo así únicamente dicha zona (ver figura 4.10).

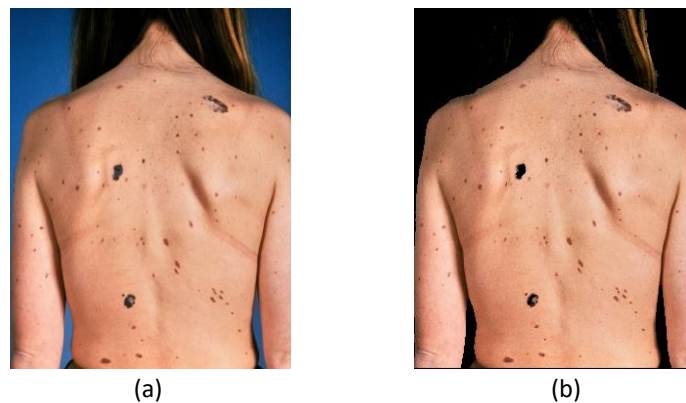


Figura 4.10: Ejemplo de resultado del segmentador basado en umbral (de color). Imagen (a) a segmentar, imagen (b) resultado de la segmentación. Imagen (a) extraída de [18]

El principal problema de este segmentador es que, si hay alguna lesión, mancha, tatuaje o zona que no tenga un color que según la aplicación considere piel, también lo va a eliminar de la imagen, por eso, se hace necesario el siguiente punto para eliminar dicho problema (ver figura 4.11).



Figura 4.11: Zonas con falsas segmentaciones

### **4.3.3 Detector de contornos**

Una vez llegados a este paso nos encontramos con dos problemas fundamentales derivados de la segmentación:

- El primero reside en que el segmentador aplicado, explicado en profundidad en el punto 4.3.2, devuelve las zonas que coinciden con el color de piel por lo que, si una persona tiene un tatuaje, mancha, lesión ... que no coincida con el color de la piel, este será eliminado de la misma manera que el resto de los objetos, lo cual impedirá hacer un análisis posterior de estas zonas, que son precisamente el objetivo final del algoritmo.
- El segundo problema que nos encontramos tiene relación con la detección de puntos característicos necesaria para hacer la homografía. Estos puntos son los que utilizará el programa para variar la perspectiva de las fotografías para que ambas coincidan, y de este modo poder compararlas. No obstante, el segmentador elimina todo aquello que no se considere piel, pero, este no es perfecto, y cabe la posibilidad que también elimine pequeñas zonas correspondientes al contorno del cuerpo obteniendo como resultado, un contorno que no es el mismo que hay en la imagen original. Dado que este contorno que obtenemos ya no es el correspondiente al de la imagen original ya que puede tener cambios, debemos forzar al algoritmo que descarte todos aquellos puntos que se encuentren en dicha zona.

Este punto del trabajo da solución a los dos problemas mencionados anteriormente. En primer lugar, este paso recoge la imagen resultante de la segmentación, y con ella, detectamos el contorno de la zona de interés.

Una vez hemos detectado el contorno ya sabemos que zona tenemos que extraer de la imagen original para evitar el primer problema citado anteriormente, obteniendo la imagen sin las falsas segmentaciones (ver Figura 4.12 - d).

Como solución al segundo problema, guardamos una copia del contorno con el cual generaremos una máscara que nos permita localizar los puntos válidos y descartar aquellos que no lo son. Esta solución se explicará con mayor detenimiento en el apartado 4.3.4.



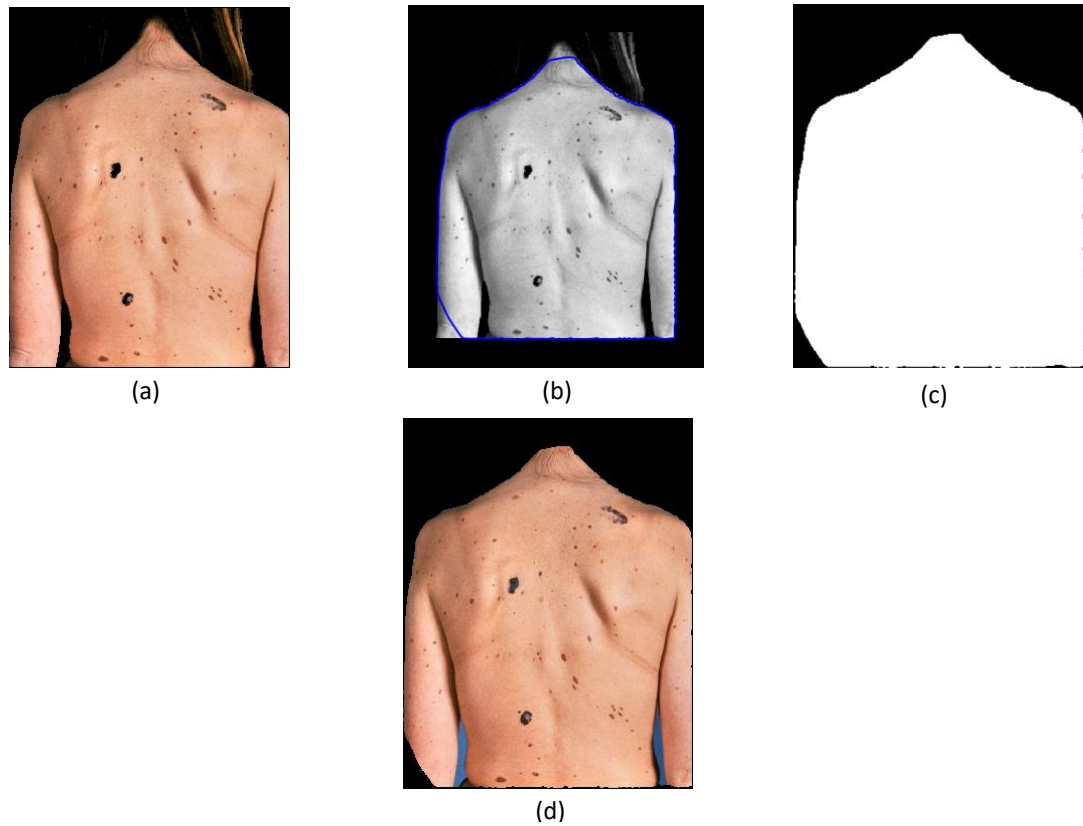


Figura 4.12: (a) Imagen segmentada. (b) contorno de la imagen (a). (c) máscara donde se detectarán los puntos característicos. (d) Imagen resultado de esta fase.

#### 4.3.4 Homografía

En este paso, tratamos con las dos imágenes que queremos comparar una vez han sido procesadas como se ha explicado en los puntos anteriores. En primer lugar, serán necesarias las dos imágenes junto con sus respectivas zonas donde queremos hallar los puntos característicos (ver Figura 4.13). Una vez tenemos estos cuatro parámetros buscamos todos los puntos característicos en cada imagen por separado y seleccionamos aquellos que se encuentran dentro de la zona de búsqueda. Podemos definir como puntos característicos de una imagen a aquellos puntos que contienen una propiedad que los definen como pueden ser esquinas, bordes, cambios de color, en definitiva, elementos que presenten alguna particularidad en la imagen. Al obtener estos puntos de ambas imágenes buscamos aquellos puntos que coincidan, los cuales nos servirán de referencia para buscar la transformación que nos permita cambiar la perspectiva de la segunda imagen, es decir, conseguiremos que las imágenes se vean como si se hubieran tomado desde el mismo punto sin haber variado tampoco el ángulo de la cámara en la que fue tomada (ver Figura 4.14). Por último, el algoritmo nos devolverá la segunda imagen con todos los cambios que hayan sido necesarios para que esta se asemeje espacialmente a la primera.

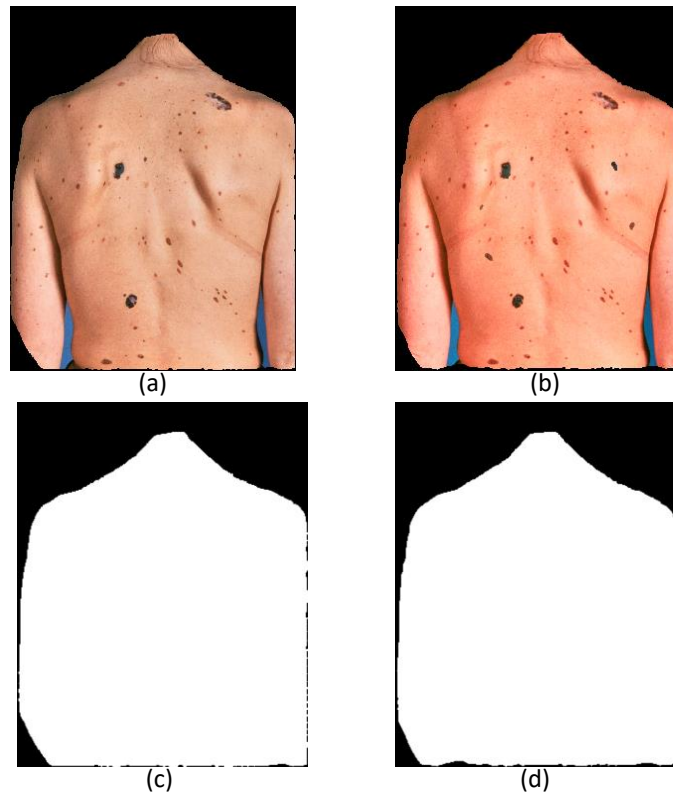


Figura 4.13: Imágenes a comparar (a) y (b) junto con sus respectivas zonas de búsqueda de puntos característicos (c) y (d)

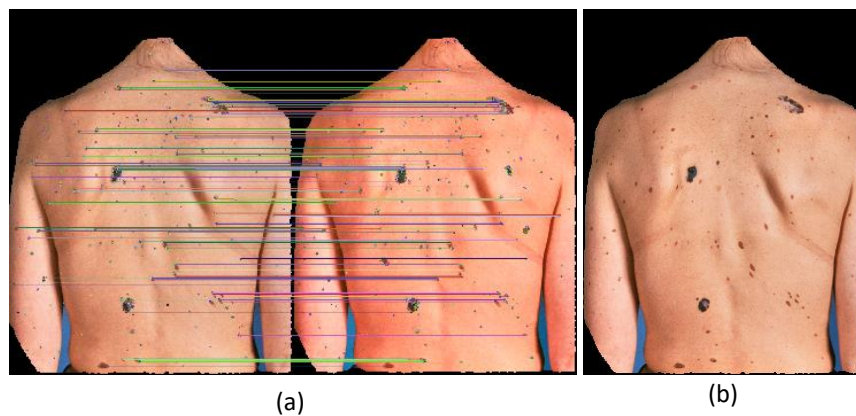


Figura 4.14: Imagen (a) donde se representan los puntos característicos de ambas imágenes y relación entre los mismos. Imagen (b) resultado de la homografía

#### 4.3.5 Igualación de color

Realizados los pasos anteriores entramos en el último paso del tratamiento de la imagen. Este paso es necesario ya que, como veremos en el proceso de comparación, ambas imágenes deben tener el color lo más parecido posible. Esto se debe a que en el momento en el que se toman las fotografías puede haber

variaciones en el tono de la piel capturado por la cámara debido a variaciones de luminosidad en la consulta, o cambios en el tono de la piel del propio paciente.

Para entender mejor este paso debemos recordar que una imagen a color se compone de tres canales (RGB) los cuales, unidos entre sí, forman los colores de la imagen. Debido al cambio de color que se ha podido producir entre ambas imágenes, el valor de estos canales habrá variado para representar el citado cambio de color. Para compensar esta variación lo que se realiza es seleccionar cada uno de los tres canales de forma individual de la imagen original, mirando cual es el valor medio de cada uno. Posteriormente se realiza lo mismo con la otra imagen, y la diferencia entre las medias de cada canal por separado será lo que haya variado cada uno de una imagen con respecto a la otra. Por último, compensamos esta diferencia existente entre los canales de la imagen, obteniendo así la imagen que podremos comparar de una manera más exacta ya que no habrá tanta variación de color entre ambas como se puede ver en la figura 4.15.

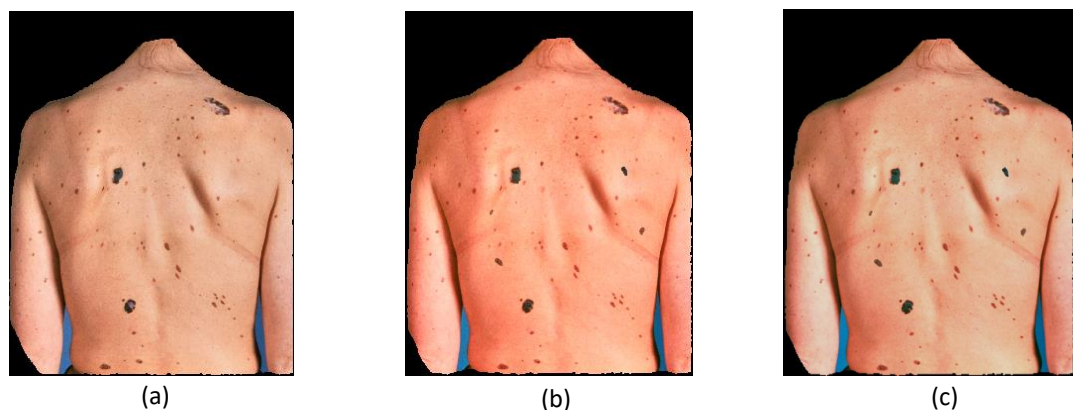


Figura 4.15: Imágenes de entrada para la ecualización de color (a) y (b). Imagen resultada (c) correspondiente de variar los canales RGB de (b) con respecto a (a)

#### **4.3.6 Comparación**

En este paso se lleva a cabo la comparación propiamente dicha donde comparamos las imágenes que han sido tratadas como se ha explicado en los apartados anteriores.

Con ambas imágenes ya tratadas de una manera que las podamos comparar de la mejor manera posible, buscamos las diferencias entre ambas. Este proceso consiste en restar las imágenes. No olvidemos que las imágenes se componen de canales, y estos canales no dejan de ser más que valores numéricos los cuales se pueden realizar operaciones con ellos. Al restar ambas imágenes conseguimos extraer aquellas zonas donde hay cambios entre una imagen y otra, siendo el resultado igual a cero en aquellas zonas donde la imagen no ha variado entre una sesión y otra, y distinta de cero si, por el contrario, ha habido cambios (ver Figura 4.16). La

comparación cabe decir, que no solo es sensible a la aparición de nuevas lesiones de la piel, sino que también es sensible al crecimiento y a la variación de color de estas.

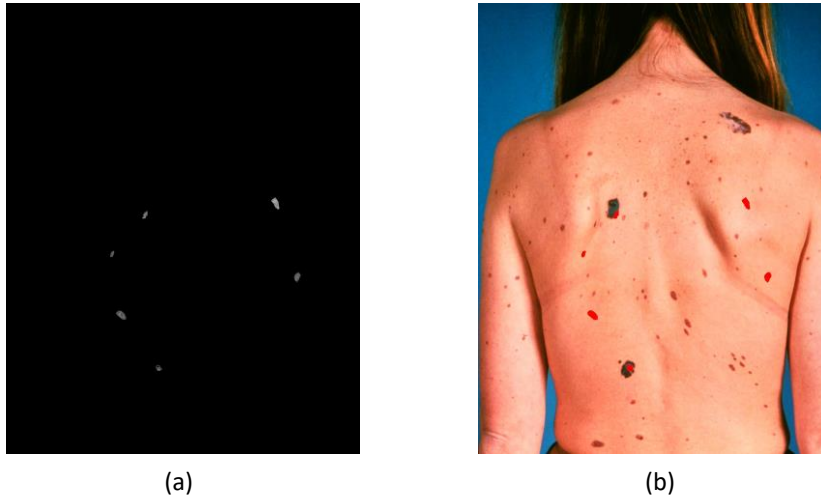


Figura 4.16: Resultado de la diferencia entre las dos imágenes a comparar (a). Representación de dichas diferencias en la imagen (b).

Una vez dicho esto, tenemos que destacar qué consideramos cambio en las imágenes y qué no. Este comparador de imágenes detecta todas las diferencias que hay entre dos imágenes, ya hayan aparecido o desaparecido manchas, crecido o menguado o cambiado de color. Como bien se ha explicado anteriormente, al restar dos imágenes se nos mostraran todas las diferencias sean de la índole que sean debido al cambio de color de esas zonas, mientras que, aquellas zonas que no hayan sufrido variación entre las dos imágenes su color apenas habrá variado, por tanto, realmente lo que consideramos diferencia son aquellas zonas que hayan sufrido una variación de color de una sesión a la siguiente.

Por último, en esta etapa hay que tener en cuenta dos ajustes para que la visualización de resultados sea lo más fiel posible. El primero de ellos es la sensibilidad en los bordes de la zona de interés y el segundo será la sensibilidad del detector de cambio. Su influencia y los errores en los que se puede incurrir si no se ajustan adecuadamente se explicará en el apartado 4.4.

#### **4.3.7 Visualización de resultados**

En este último paso del algoritmo generamos la imagen que se le mostrará al usuario para que pueda visualizar de una manera clara e intuitiva dónde se han producido los cambios en la piel.

Una vez tenemos las diferencias de las dos imágenes que hemos comparado las representamos en la imagen, resaltando de color rojo el área de la piel que ha variado como se puede ver en la figura 4.16 - b. Sin embargo, esta área puede ser

muy pequeña, por lo que el usuario podría no darse cuenta de ciertos cambios en la imagen, por tanto, se instauro un mecanismo en el que todas aquellas zonas donde se hayan detectado cambios se encuadren para que el usuario, aunque no vea un cambio a simple vista porque el área de cambio sea muy pequeña, sí que localice a simple vista las zonas donde la imagen ha variado (ver figura 4.17).



Figura 4.17: Ejemplo de imagen donde se muestran los cambios encuadrados.

Por último, el usuario obtendrá una imagen compuesta por las dos imágenes que han sido comparadas y una tercera donde se mostraran todos aquellos cambios detectados por el algoritmo (ver figura 4.18).



Figura 4.18: Ejemplo de resultado que obtendría el usuario del algoritmo.

#### **4.4 Parámetros de diseño del algoritmo**

En este apartado se van a mostrar los errores que puede tener la aplicación en función de los dos parámetros fundamentales que se pueden variar en el mismo. Hay que tener en cuenta que el usuario, como se ha visto en el apartado anterior, no obtendrá los resultados sobre la imagen segmentada, sin embargo, para la redacción de este



apartado, sí que se mostraran en esta imagen para que se vea de manera más clara la influencia de los dos parámetros que se van a explicar a continuación:

- 1- El primer parámetro que podemos variar es la sensibilidad en los bordes de la zona de interés. Si recordamos en el punto 4.2.3, obtenemos el contorno de la imagen para así seleccionar de la imagen original nuestra zona de interés sin posibles falsas segmentaciones. Este contorno en las imágenes que se quieren comparar no tiene por qué ser necesariamente el mismo por lo que, al restar ambas imágenes saldrán amplias diferencias en los bordes de la imagen (ver figura 4.19).

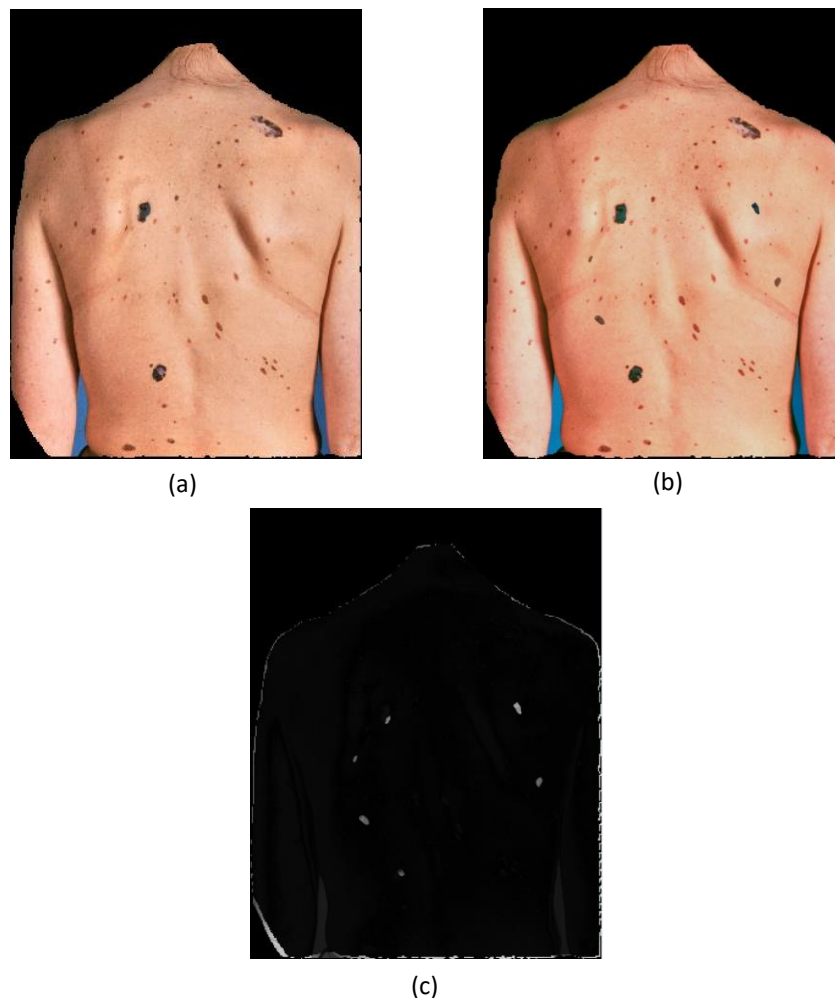


Figura 4.19: Imágenes a las que se les haya la diferencia (a) y (b). Imagen diferencia (c) donde podemos ver las diferencias notables que surgen en el contorno del cuerpo.

Este problema se soluciona variando un parámetro que elimina un porcentaje del contorno de interés de la imagen, que, cuanto más agresivo sea, se corre el riesgo de eliminar también zonas de interés cercanas a dichos bordes donde puede haber nuevas lesiones o variaciones de las existentes. Veamos algunos ejemplos:

- Reducción del **0,01%** en la zona de interés

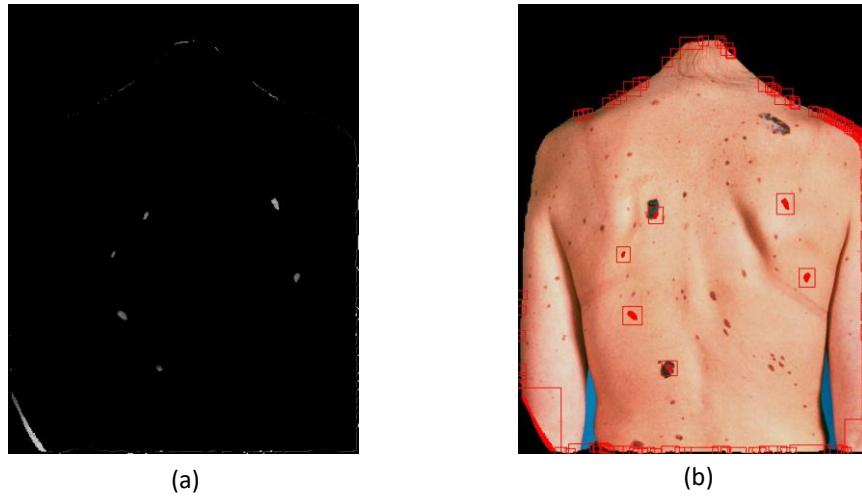


Figura 4.20: Imagen (a) resultado de la diferencia entre imágenes con una reducción de la zona de interés del 0,01%. Imagen (b) resultado de la comparación entre las imágenes (a) y (b) de la figura 4.19 con dicha reducción de borde.

Como podemos ver, con una reducción mínima se detectan gran cantidad de falsos positivos en el límite de la zona de interés (ver figura 4.20).

- Reducción **0,1%** de la zona de interés

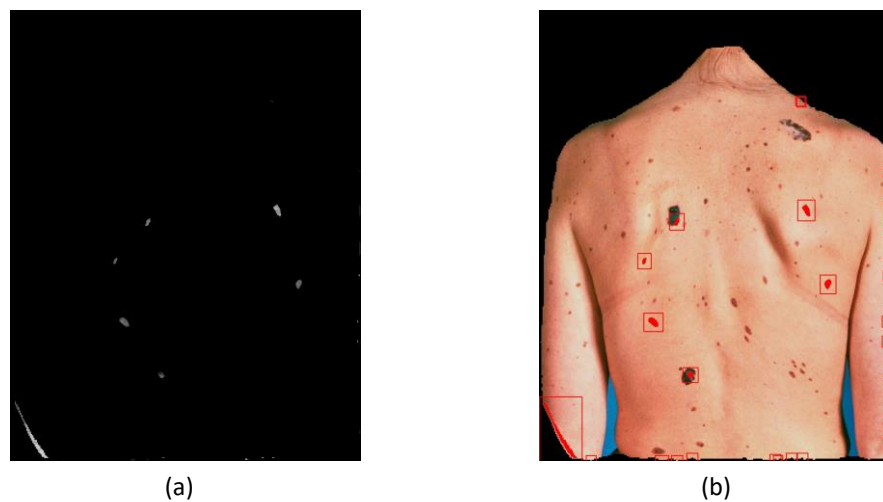


Figura 4.21: Imagen (a) resultado de la diferencia entre imágenes con una reducción de la zona de interés del 0,1%. Imagen (b) resultado de la comparación entre las imágenes (a) y (b) de la figura 4.19 con dicha reducción de borde.

Con una reducción del 0,1% de la zona de interés vemos como cada vez se observan menos falsos positivos en los bordes de la zona de interés (ver figura 4.21).

- Reducción del **0,3%** de la zona de interés

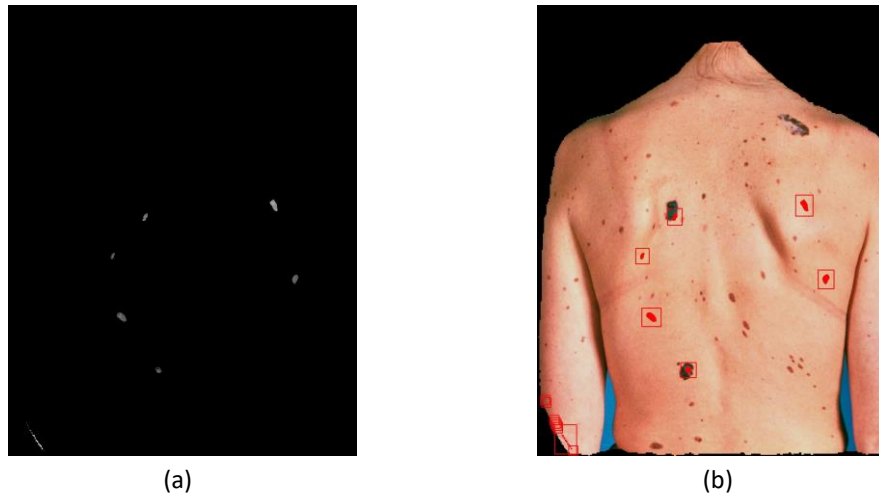


Figura 4.22: Imagen (a) resultado de la diferencia entre imágenes con una reducción de la zona de interés del 0,3%. Imagen (b) resultado de la comparación entre las imágenes (a) y (b) de la figura 4.19 con dicha reducción de borde.

Con una reducción del 0,3% vemos como surgen únicamente falsos positivos en la zona inferior izquierda de la imagen (b) de la figura 4.22.

- Reducción del **0,5%** en la zona de interés

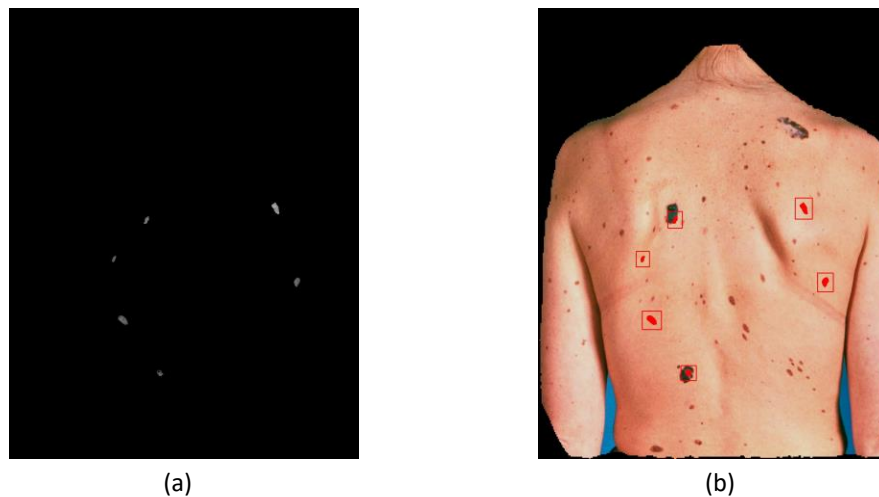


Figura 4.23: Imagen (a) resultado de la diferencia entre imágenes con una reducción de la zona de interés del 0,5%. Imagen (b) resultado de la comparación entre las imágenes (a) y (b) de la figura 4.19 con dicha reducción de borde.

Como podemos ver en la figura 4.23 no se detectan falsos positivos en los bordes de la región de interés.



2- El segundo parámetro que se puede variar una vez ajustado el primero es la sensibilidad del detector. Este parámetro lo que regula es la sensibilidad que tendrá el detector a la hora de localizar diferencias entre las imágenes. Como bien hemos dicho en el apartado 4.3.5, entre las imágenes que vamos a comparar pueden existir diferencias en cuanto al color de la imagen se refiere que, aunque se intentan compensar en dicho apartado, no se puede llegar a conseguir la igualdad absoluta en el color de la piel por lo que, al restar ambas imágenes, existirán a lo largo de la misma mínimas diferencias que el detector mostrará como cambios en la piel. Dado que el color en las zonas que no han variado será prácticamente el mismo, las diferencias serán muy bajas con respecto a las lesiones que realmente han variado en una imagen con respecto a la otra. Esta sensibilidad se mide de la siguiente manera: Con una sensibilidad total, la aplicación detectará todos aquellos puntos donde la diferencia de imágenes sea mayor a 0 entre una y otra, a medida que vaya aumentando esta diferencia se detectarán aquellos puntos que cumplan con dicha diferencia como mínimo, siendo la máxima diferencia posible entre imágenes de 255. A continuación, vamos a ver algunos ejemplos sobre cómo influye este parámetro en el detector:

- Sensibilidad **1**

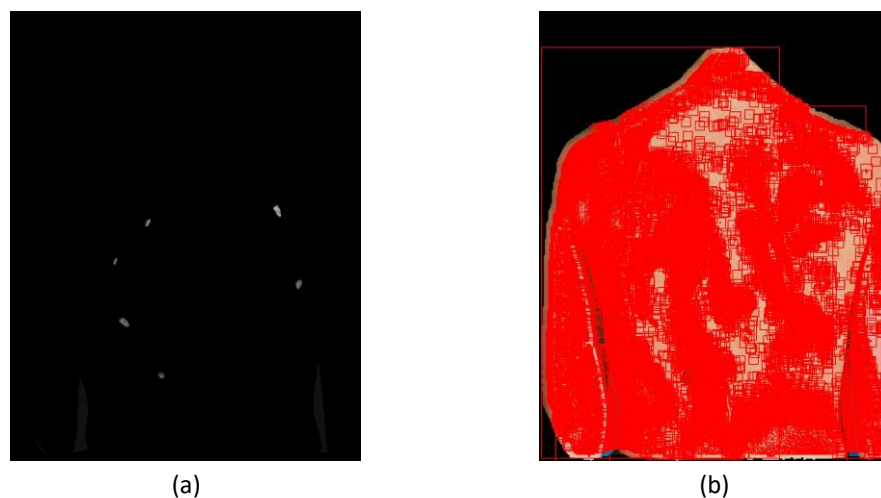


Figura 4.24: Imagen (a) resultado de la diferencia entre imágenes con una sensibilidad de 1 y la diferencia en el contorno eliminada. Imagen (b) resultado de la comparación entre las imágenes (a) y (b) de la figura 4.19 con dicha sensibilidad.

Como podemos ver con una sensibilidad de 1, la aplicación detecta diferencias prácticamente en toda la zona de interés siendo esta sensibilidad muy poco útil para la detección de nuevas lesiones (ver figura 4.24).

- **Sensibilidad 10**

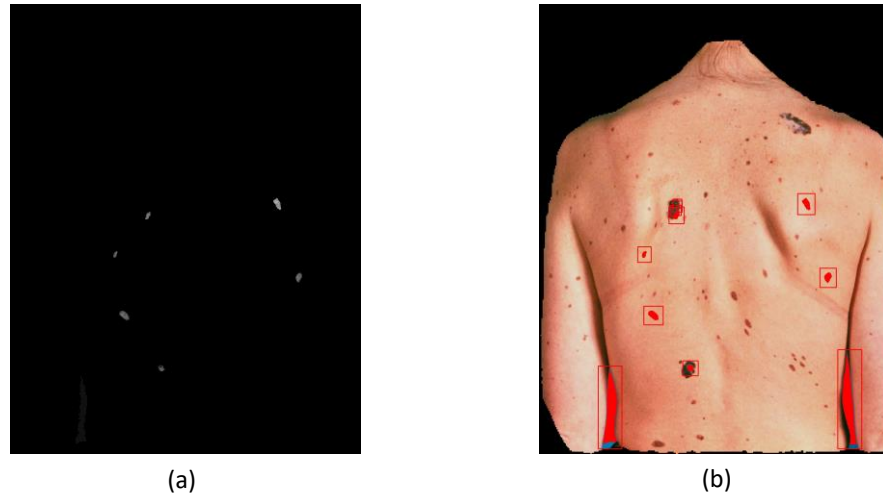


Figura 4.25: Imagen (a) resultado de la diferencia entre imágenes con una sensibilidad de 10 y la diferencia en el contorno eliminada. Imagen (b) resultado de la comparación entre las imágenes (a) y (b) de la figura 4.19 con dicha sensibilidad.

Con una sensibilidad igual a 10, observamos que las diferencias de aquellas zonas que no han variado eran bastante pequeñas, y solo ha habido que aumentar ligeramente la sensibilidad para que sea posible ver de una forma clara donde se encuentran las lesiones (ver figura 4.25).

- **Sensibilidad 20**

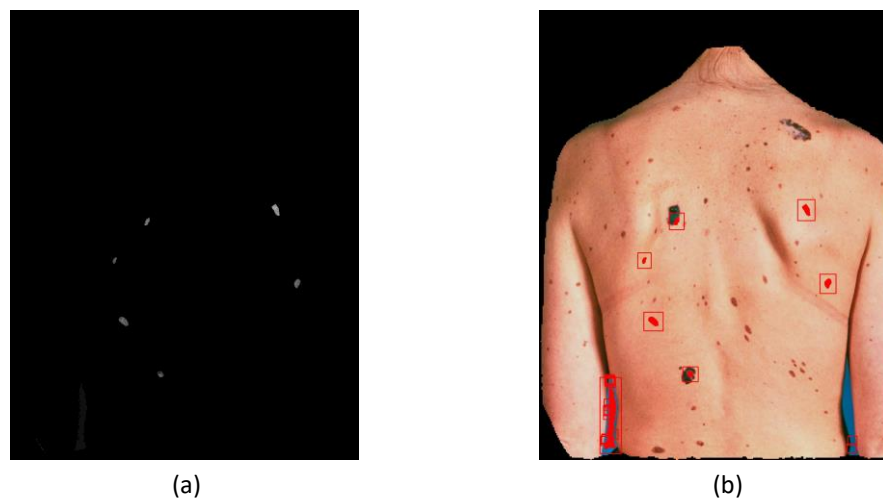


Figura 4.26: Imagen (a) resultado de la diferencia entre imágenes con una sensibilidad de 20 y la diferencia en el contorno eliminada. Imagen (b) resultado de la comparación entre las imágenes (a) y (b) de la figura 4.19 con dicha sensibilidad.

Con una sensibilidad de 20, la aplicación ha eliminado prácticamente por completo la detección de lesiones falsas (ver figura 4.26).

- Sensibilidad **30**

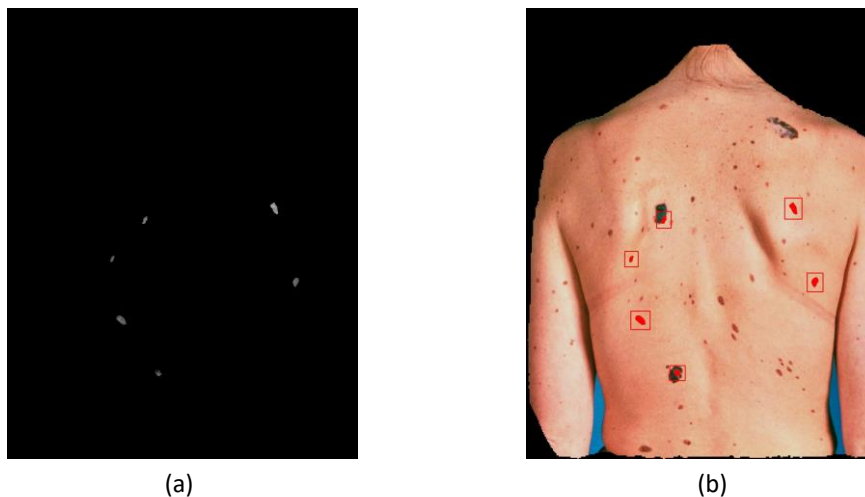


Figura 4.27: Imagen (a) resultado de la diferencia entre imágenes con una sensibilidad de 1 y la diferencia en el contorno eliminada. Imagen (b) resultado de la comparación entre las imágenes (a) y (b) de la figura 4.19 con dicha sensibilidad.

Con una sensibilidad de 30, se ha eliminado por completo aquellas detecciones que no son útiles manteniendo solo aquellas que son relevantes.

Hay que clarificar que en función de las fotografías y de la diferencia resultante entre ambas, así como lo preciso que haya sido el detector de contornos explicado en 4.3.3, los parámetros óptimos podrían variar entre imágenes. En el siguiente ejemplo se ve como los parámetros óptimos para la imagen anterior (reducción del 0,5% y sensibilidad de 30) no son lo suficientemente óptimos para la que se presenta a continuación:



Figura 4.28: Imágenes a comparar y resultado con sensibilidad 30 y reducción de la zona de interés del 0,5%

Como conclusión a este apartado, el usuario debería poder ajustar la sensibilidad del algoritmo para adaptar a cada imagen concreta el detector de diferencias. Sin embargo, se propusieron unos valores que aportan muy buenos resultados, como

se podrá ver en el Capítulo 5 para que, el usuario, tenga la menor implicación a nivel de parámetros y hacer la ejecución del algoritmo algo sencillo.

## 5 Pruebas y resultados

---

### 5.1 Dataset

El objetivo inicial de este trabajo era aplicar el algoritmo diseñado sobre un Dataset (conjunto de elementos que se utilizan para evaluar un algoritmo) de imágenes obtenido de una colaboración con el Hospital Universitario Puerta de Hierro de Madrid (HUPHM). Sin embargo, al coincidir la etapa de anonimización y adaptación de dichas imágenes (marzo de 2020) con el comienzo de las medidas de confinamiento derivadas de la COVID-19, no ha sido posible llevarla a cabo y ha sido necesario buscar una alternativa.

El Dataset utilizado finalmente en este TFG ha sido extraído de internet. Para la confección del Dataset se han seleccionado diez imágenes de personas con lesiones dermatológicas en la zona de la espalda donde, añadiendo, ampliando o variando lesiones se ha podido proseguir con el trabajo y comprobar la eficacia del algoritmo diseñado. Estas lesiones se han añadido utilizando o modificando fragmentos de las lesiones ya existentes en las imágenes, para hacerlas lo más realistas posible. Por otro lado, se han seleccionado imágenes que nos permitieran ver cómo se comporta el algoritmo en situaciones no ideales y ver en qué circunstancias se podría mejorar el mismo, así como ver en qué lugares falla.

### 5.2 Resultados visuales

A continuación, se muestran los resultados visuales extraídos del *Dataset* que obtendría un usuario al ejecutar la aplicación. Estos resultados se han obtenido con una sensibilidad de **25** y una reducción de la zona de interés del **0,4%**. De todas las pruebas realizadas, con estos valores son con los que obtenemos mejores resultados y serán la configuración por defecto que tendría la aplicación. Si bien se obtienen unos resultados muy buenos, hay ciertas detecciones que no se realizan de la manera óptima (ver Figura 5.1 imágenes (a) e (i)) detectando zonas que no son de interés para el usuario, lo cual no impide que detecte las nuevas lesiones que han surgido.



(a)



(b)



(c)



(d)





(e)



(f)



(g)



(h)



(i)



(j)

Figura 5.1: Imágenes resultantes de la prueba del algoritmo con una sensibilidad de 25 y una reducción de la zona de interés del 0,4%.



## 6 Conclusiones y trabajo futuro

---

### 6.1 Conclusiones

La finalidad de este trabajo es el desarrollo de un algoritmo el cual permita el análisis comparativo de dos imágenes, distanciadas en el tiempo, de zonas amplias del cuerpo con el objeto de hallar y resaltar las diferencias encontradas de una forma clara y sencilla al usuario.

Para ello se realizó un procesamiento de imágenes, donde, en primer lugar, se implementó un segmentador de color el cual extrae todas aquellas zonas que coinciden con el color de la piel y eliminando todas aquellas zonas que no son de interés para el análisis.

Una vez que el segmentador aísla la región de interés, se procede a extraer la misma, pero de la imagen original con el fin, como se detalló en el punto 4.3.2, de evitar falsas segmentaciones que podrían dar lugar a una comparación errónea.

El siguiente paso, aplicado sobre las imágenes aisladas, consiste en realizar la homografía e igualación de color para conseguir la máxima homogenización entre imágenes tanto en su posición de captura como en el color de piel.

Por último, se eliminaron las partes comunes de las imágenes con el fin de extraer las diferencias existentes entre ambas imágenes y, una vez tenemos dicha información, se muestra al usuario de una manera clara sobre la imagen original.

El algoritmo desarrollado puede tener aplicación tanto en la detección de nuevas lesiones como en el seguimiento de otras ya existentes, automatizando así el proceso y ayudando a reducir los errores de apreciación en cuanto a tamaño y/o tono de la lesión realizada por el ojo humano.

Hemos podido comprobar también que, a la vista de los resultados, este algoritmo tiene un funcionamiento bastante preciso donde, en caso de haber error en la detección, este, es aditivo, es decir, se detectan nuevas lesiones en zonas donde realmente no está presente una nueva lesión. No obstante, como se pudo ver en el apartado 4.4, este problema se puede solventar ajustando correctamente los parámetros de sensibilidad.

Como conclusión, cabría destacar que este trabajo puede dar pie a una mejora sustancial en cuanto al tiempo que emplea el ser humano en comparar las imágenes, así como una reducción considerable en la posibilidad de fallo en detección del cambio, ya sea porque haya cambios prácticamente inapreciables o bien porque la persona que compare las imágenes haya podido pasar por alto algún detalle.

## **6.2 Trabajo futuro**

Con los resultados obtenidos y evaluando la implementación del algoritmo se proponen los siguientes aspectos para que el algoritmo sea totalmente útil para un usuario medio y, por otro lado, que este sea mejor en cuanto a gestión de recursos se refiere:

- Generar una interfaz gráfica donde el usuario podría cargar las imágenes sin tener que buscar el mismo las rutas de los ficheros e introducirlas en el código.
- Generar un sistema online para que el usuario pueda tener las fotografías de cada paciente almacenadas en una cuenta personal en un servidor, sin necesidad de tenerlas en su dispositivo personal.
- Que el usuario sea capaz de seleccionar aquella región que quiere comparar entre ambas imágenes. Esto permitiría eliminar las etapas del segmentador y la detección de contornos ya que quedarían definidas por el usuario las regiones de comparación.
- Permitir al usuario variar los parámetros de sensibilidad del algoritmo. En esta primera implementación del algoritmo esta variación no está permitida para el usuario ya que, el objetivo de este algoritmo es que sea utilizado por alguien que no necesariamente sepa sobre el funcionamiento interno del algoritmo.

## Referencias

---

- [1] Adysantana. (2015). How To Get Rid Of Hormonal Acne. *How To Get Rid Of Hormonal Acne*. Recuperado de <https://howtogetridofhormonalacne.wordpress.com/2015/04/11/how-to-get-rid-of-hormonal-acne/>
- [2] Akbaş, A., Kiliç, F., Şener, S., Süngü, N., Kiliçaslan, A., y Metin, A. (2016). Primary Cutaneous Follicle Center Lymphoma: A Case Report. *Our Dermatology Online* 7 (2):181-84. DOI: 10.7241
- [3] Anar Mikailov, Adam Blechman. Gigapixel Photography for Skin Cancer Surveillance: A Novel Alternative to Total-Body Photography, 2013, 92: 241-243.
- [4] Callieri, M., Scopigno, R., Cignoni, P., Gaggio, G., Romanelli, M., Coluccia, M. y Pingi, P. (2003). *Derma: monitoring the evolution of skin lesions with a 3D system*. 167-174.
- [5] Dana Sokolová, N. (2013). Varovné signály, jež mohou včas upozornit na rakovinu kůže. *Novinky.cz*. Recuperado de <https://www.novinky.cz/zena/zdravi/clanek/varovne-signal-y-jez-mohou-vcas-upozornit-na-rakovinu-kuze-191916>.
- [6] Dr. J. Ramiro Martínez de Dios. Técnicas de Segmentación. Máster en Automática, Robótica y Telemática, 2016.
- [7] Eliseu C. Martinho (2016). *Oh my Exif!* Recuperado de <https://medium.com/@Orangeeli/oh-my-exif-777335e26213>
- [8] Escudero Gonzales, J. P. (2018). *Análisis de texturas en imágenes de tomografía computerizada para la caracterización de tumores pulmonares atendiendo a su naturaleza* (Trabajo de fin de grado). Universidad Politécnica de Valencia. Valencia, España. Recuperado de <https://riunet.upv.es/handle/10251/104945>
- [9] Gerdler, Y. (2019). La varicela puede tener graves complicaciones sin el debido cuidado. *Televen*. Recuperado de <http://www.televen.com/elnoticiero/bienestar/la-varicela-puede-graves-complicaciones-sin-debido-cuidado/>
- [10] Instituto Dermatológico de Felipe. (2017). Tratamiento eliminar manchas en Barcelona. *Clínica de Felipe*. Recuperado de <https://www.defelipe.com/manchas/>

- [11] Karami, Ebrahim, Siva Prasad, y Mohamed Shehata. (2017). Image Matching Using SIFT, SURF, BRIEF and ORB: *Performance Comparison for Distorted Images*. Faculty of Engineering and Applied Sciences. Memorial University, Canada.
- [12] Kolkur, S., D. Kalbande, P. Shimpi, C. Bapat, y J. Jatakia. (2017). Human Skin Detection Using RGB, HSV and YCbCr Color Models. *Proceedings of the International Conference on Communication and Signal Processing* (137), 324-332.
- [13] M. Callieri, P. Cignoni, P. Pingi, R. Scopigno, M. Coluccia, G. Gaggio, M. Romanelli. *Derma: monitoring the evolution of skin lesions with a 3D system*, 2003.
- [14] M. Martí, Rosa. (2012). *Nevus clínicamente atípicos y síndrome del nevus displásico*. Dermatoweb. Recuperado de <http://dermatoweb.udl.es/seminarios.php?relacio=1&idsubmenu=2651&idapartat=3821&idsubapartat=>
- [15] Marshal, Hanan. (2018). Natural Cures for Skin Rash - Coriander, Neem and Olive Oil. *Collection of Best Home Remedies and Natural Treatments in USA*. Recuperado de <https://www.homenaturalcures.com/rash-treatment-home-remedy-cure/>
- [16] Martinez de Dios, J. R. Visión por computador. *Técnicas de segmentación I*. Recuperado de [http://www.esi2.us.es/~jdedios/asignaturas/Master\\_1.pdf](http://www.esi2.us.es/~jdedios/asignaturas/Master_1.pdf)
- [17] Mikailov, A. y Blechman, A. (2013). Gigapixel Photography for Skin Cancer Surveillance: A Novel Alternative to Total-Body Photography. (92), 241-243
- [18] O'Connor, B., Gibson, G. y McKenna, A. (2019). *How to spot an early skin cancer and management of premalignant skin conditions*. Recuperado de <https://www.bonsecours.ie/gp-study-day-presentations-2019?STAFFFOLDER=&CANONICAL=&documentId=500>
- [19] Ortiz Cirugeda, J. M. (2014). *Aplicación android para la obtención de información a partir de fotografías*. (Trabajo de fin de grado). Universidad Autónoma de Madrid. Madrid, España. Recuperado de <https://repositorio.uam.es/handle/10486/662311>
- [20] Ramírez, J. A. y Chacón, M. I. (2011). Redes neuronales artificiales para el procesamiento de imágenes, una revisión de la última década. *Revista de ingeniería eléctrica, electrónica y computación*, (9), 7-16.

- [21] Revathi V.L, Chithra A.S. A Review on Segmentation Techniques in Skin Lesion. *International Research Journal of Engineering and Technology*, 2015: 2598 -2602
- [22] Revathi, I. y Chithra (2016). A Review on Segmentation Techniques in Skin Lesion Images. *International Research Journal of Engineering and Technology (IRJET)*. (2), 2598-2603.
- [23] Roberto Velert, C. (2016). *Análisis de imágenes de tomografía axial computerizada cardíaca para la segmentación automática de grandes trabéculas*. (Trabajo de fin de grado). Universidad Politécnica de Valencia. Valencia, España. Recuperado de <https://riunet.upv.es/handle/10251/70104>
- [24] Rodriguez Cancio, M. (2014). *Aplicaciones para la segmentación de imágenes realizadas en el Centro de Investigación de Métodos Computacionales y Numéricos en la Ingeniería de la UCLV en el período 2011-2013*. (Máster en señales y sistemas). Universidad Central “Marta Abreu” de Las Villas. Santa Clara, Cuba. Recuperado de <https://dspace.uclv.edu.cu/handle/123456789/6937>
- [25] S. Kolkur, D. Kalbande, P. Shimpi, C. Bapat, J. Jatakia. Human Skin Detection Using RGB, HSV and YCbCr Color Models, 2017: 324 – 332.
- [26] Scikit-image, image processing in python. (2020). *Comparing edge-based and region-based segmentation — skimage v0.18.dev0 docs*. Recuperado de [https://scikit-image.org/docs/dev/auto\\_examples/applications/plot\\_coins\\_segmentation.html](https://scikit-image.org/docs/dev/auto_examples/applications/plot_coins_segmentation.html)
- [27] Smitt J., van Everdingen J., Starink T., van der Horst H. (2014) *14 Benigne huidtumoren*. In: *Dermatovenereologie voor de eerste lijn*. Bohn Stafleu van Loghum, Houten
- [28] Stücker, C. (2009). *UV-Strahlung als berufsbedingter Risikofaktor am Beispiel von Bauarbeitern*. (Trabajo de fin de grado). Medizinischen Universität Graz. Auenbruggerpl, Austria. Recuperado de [https://online.medunigraz.at/mug\\_online/wbAbs.showThesis?pThesisNr=16232&pOrgNr=1&pPersNr=51711](https://online.medunigraz.at/mug_online/wbAbs.showThesis?pThesisNr=16232&pOrgNr=1&pPersNr=51711)
- [29] Valverde-Rebaza, j. (2007). Figura 3. Resultado de La Detección de Bordes Mediante El Algoritmo De... *ResearchGate*. Recuperado de [https://www.researchgate.net/figure/Figura-3-Resultado-de-la-deteccion-de-bordes-mediante-el-algoritmo-de-Canny\\_fig2\\_267240432](https://www.researchgate.net/figure/Figura-3-Resultado-de-la-deteccion-de-bordes-mediante-el-algoritmo-de-Canny_fig2_267240432)

- [30] Xin-Yi Gong, Hu Su, De Xu, Zheng-Tao Zhang, Fei Shen, Hua-Bin Yang. An Overview of Contour Detection Approaches[J]. International Journal of Automation and Computing, 2018, 15(6): 656-672